

For Reference

NOT TO BE TAKEN FROM THIS ROOM

Ex LIBRIS
UNIVERSITATIS
ALBERTAENSIS



THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR James C. Mozel

TITLE OF THESIS Parameter Estimation Using The University of
Alberta Identification Program

DEGREE FOR WHICH THESIS WAS PRESENTED M.Sc.

YEAR THIS DEGREE GRANTED 1980 Fall

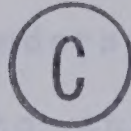
Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

THE UNIVERSITY OF ALBERTA

PARAMETER ESTIMATION USING THE
UNIVERSITY OF ALBERTA IDENTIFICATION PROGRAM

by



JAMES C. MOZEL

A THESIS SUBMITTED

TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF SCIENCE
IN
CHEMICAL ENGINEERING

DEPARTMENT OF CHEMICAL ENGINEERING

EDMONTON, ALBERTA

FALL, 1980

THE UNIVERSITY OF ALBERTA
FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read,
and recommend to the Faculty of Graduate Studies and
Research, for acceptance, a thesis entitled PARAMETER
ESTIMATION USING THE UNIVERSITY OF ALBERTA
IDENTIFICATION PROGRAM submitted by JAMES C. MOZEL in
partial fulfilment of the requirements for the degree of
MASTER OF SCIENCE in CHEMICAL ENGINEERING.

ACKNOWLEDGMENTS

The author gratefully acknowledges the assistance, encouragement and patience provided by his thesis advisor, Dr. R.K. Wood.

Thanks are due to the staff of the Data Acquisition, Control and Simulation Centre of the Department of Chemical and Petroleum Engineering and Mr. Robert Bilec for their assistance.

Finally, the author wishes to express his appreciation to Alanna Mozel, who typed the manuscript of this thesis.

Financial support from the University of Alberta is gratefully acknowledged.

ABSTRACT

The four estimation algorithms in the University of Alberta Identification Package were tested using computer generated and experimental data. A double pipe countercurrent liquid-liquid heat exchanger and a binary distillation column with eight bubble cap trays were the two experimental units used in this study. An IBM 1800 digital computer, executing Direct Digital Control (DDC), was used to control the units and to accumulate the data.

A considerable portion of this thesis is concerned with the theory of the four identification routines:

- 1) Generalized Least Squares (STIME)
- 2) Maximum Likelihood (ASTRM)
- 3) Instrumental Variables (IVOFF)
- 4) Unbiased Estimator (UBIAS)

and the software of the package. The University of Alberta's Amdahl 470V/6 digital computer was used to implement the scheme.

The results of the computer generated data identification runs demonstrate the effect of:

- 1) lengthening the input/output data sequences
on the accuracy of the parameter estimates
- 2) increasing the model order on the accuracy of
the parameter estimates
- 3) the type of identification scheme used

4) the effect of noisy data on the accuracy of the parameter estimates

These results were compared with findings from other studies and general agreement was found. The generalized least squares identification routine produced results that were unacceptable for good process modelling. The preferred identification routines were the instrumental variables and the unbiased estimator methods.

The experimental results obtained from the heat exchanger were compared to previous investigations and the models estimated for the distillation column were used by a fellow student. Some problem occurred in estimating the gain of a process with a large time constant. A solution was proposed and proved with testing, that this problem could be reduced if the clock interval of the input disturbance signal was increased. The model proposed for the heat exchanger was in good agreement with models from other investigations but some problems were encountered in estimating the time delays of the process.

TABLE OF CONTENTS

CHAPTER		PAGE
I.	Introduction	1
II.	Theory of Identification Methods . .	4
	Introduction	4
	Statement of the Problem . . .	5
	Assumptions	9
	Generalized Least Squares . . .	10
	Maximum Likelihood	21
	Instrumental Variables	30
	Unbiased Estimator	33
III.	Organization and Structure of the University of Alberta Identification Package	38
	Introduction	38
	Program Modules	42
	OXIP	42
	STEER	48
	HELP	56
	DATGET	60
	DAGEN	64
	DMOVE	74
	CRCOR	81
	STIME	86
	ASTRM	97
	IVOFF	106
	UBIAS	110
	DIAGN	117
	MDISP	124
	NOISE	129
	FINAL	134

CHAPTER		PAGE
	FDISP	139
	FDIS2	145
IV.	Parameter Estimation of a Simulated Process using UNALIP	150
	Introduction	150
	Test Cases	151
	Identification	155
	Summarized Results	173
	Discussion	176
V.	Parameter Estimation of Models for Pilot Plant Units	178
	Introduction	178
	Distillation Column	179
	Heat Exchanger	181
	Experimental Procedure	183
	Distillation Column	183
	Heat Exchanger	183
	Results	185
	Distillation Column	185
	Heat Exchanger	194
	Discussion	199
IV.	Conclusions and Recommendations	202
	Bibliography	206
	Appendix A	212
	Appendix B	216
	Appendix C	222

Appendix D	242
----------------------	-----

LIST OF TABLES

TABLE		PAGE
4.1	High Noise/Signal Ratios for each Noise Sequence and Process	160
4.2	Low Noise/Signal Ratios for each Noise Sequence and Process	160
4.3	Comparison of the Standard Deviations of the Error Measures for Generalized Least Squares (STIME)	163
4.4	Comparison of the Standard Deviations of the Error Measures for Maximum Likelihood (ASTRM)	164
4.5	Comparison of the Standard Deviations of the Error Measures for Instrumental Variables (IVOFF)	165
4.6	Comparison of the Standard Deviations of the Error Measures for Unbiased Estimator (UBIAS)	166
4.7	Comparison of the Average Standard Deviations of the Parameters Estimated by Generalized Least Squares (STIME)	167
4.8	Comparison of the Average Standard Deviations of the Parameters Estimated by Maximum Likelihood (ASTRM)	168
4.9	Comparison of the Average Standard Deviations of the Parameters Estimated by Instrumental Variables (IVOFF)	169
4.10	Comparison of the Average Standard Deviations of the Parameters Estimated by Unbiased Estimator (UBIAS)	170
4.11	Comparison of the Computational Time (seconds) Required for the Estimation Routines	172

TABLE		PAGE
5.1	Comparison of Parameters Estimated by Three Identification Routines for the Steam and Bottom Composition Transfer Function	186
5.2	Comparison of Parameters Estimated by Three Identification Routines for the Steam and Top Composition Transfer Function	187
5.3	Comparison of Parameters Estimated by Three Identification Routines for the Reflux and Bottom Composition Transfer Function	188
5.4	Comparison of Parameters Estimated by Three Identification Routines for the Reflux and Top Composition Transfer Function	189
5.5	Identified Gains of Simulated System Using Each Estimation Technique	191
5.6	Effect of Clock Interval on the Estimated Gain of a Simulated System	193

LIST OF FIGURES

FIGURE		PAGE
3-1	Flow Diagram for OXIP	43
3-2	Flow Diagram for STEER	49
3-3	Flow Diagram for STEER	50
3-4	Flow Diagram for STEER	51
3-5	Flow Diagram for HELP	57
3-6	Flow Diagram for DATGET	61
3-7	Box Diagram for DAGEN	66
3-8	Flow Diagram for DAGEN	67
3-9	Flow Diagram for DAGEN	68
3-10	Box Diagram for DMOVE	75
3-11	Flow Diagram for DMOVE	76
3-12	Flow Diagram for CRCOR	82
3-13	Flow Diagram for STIME	88
3-14	Flow Diagram for STIME	89
3-15	Flow Diagram for STIME	90
3-16	Flow Diagram for ASTRM	99
3-17	Flow Diagram for ASTRM	100
3-18	Flow Diagram for ASTRM	101
3-19	Flow Diagram for IVOFF	107
3-20	Flow Diagram for IVOFF	108
3-21	Flow Diagram for UBIAS	112
3-22	Flow Diagram for UBIAS	113
3-23	Flow Diagram for UBIAS	114
3-24	Flow Diagram for DIAGN	119
3-25	Flow Diagram for MDISP	126

FIGURE		PAGE
3-26	Flow Diagram for NOISE	131
3-27	Flow Diagram for FINAL	135
3-28	Flow Diagram for FDISP	140
3-29	Flow Diagram for FDIS2	146
4-1	General Block Diagram of Simulated Processes	153
4-2	Pseudo Random Binary Sequence Used as Input for Simulation Runs	156
5-1	Simple Schematic of the Distillation Column	180
5-2	Simple Schematic of the Heat Exchanger	182
B-1	Tube Flow Rate Response for Test A	218
B-2	Tube Flow Rate Response for Test B	219
B-3	Shell Outlet Temperature Response for Test A	220
B-4	Shell Outlet Temperature Response for Test B	221
C-1	Reflux Flow Rate Response for PRBS Disturbance of Length 63	224
C-2	Reflux Flow Rate Response for PRBS Disturbance of Length Double 31	225
C-3	Steam Flow Rate Response for PRBS Disturbance of Length 63	226
C-4	Steam Flow Rate Response for PRBS Disturbance of Length Double 31	227
C-5	Bottom Composition Response for PRBS Disturbance of Length 63 using Steam	228
C-6	Bottom Composition Response for PRBS Disturbance of Length 63 using Reflux	229

FIGURE		PAGE
C-7	Bottom Composition Response for PRBS Disturbance of Length Double 31 using Steam	230
C-8	Bottom Composition Response for PRBS Disturbance of Length Double 31 using Reflux	231
C-9	Top Composition Response for PRBS Disturbance of Length 63 using Steam	232
C-10	Top Composition Response for PRBS Disturbance of Length 63 using Reflux	233
C-11	Top Composition Response for PRBS Disturbance of Length Double 31 using Steam	234
C-12	Top Composition Response for PRBS Disturbance of Length Double 31 using Reflux	235
C-13	Reflux Flow Rate Response for PRBS Disturbance of Length 127	236
C-14	Steam Flow Rate Response for PRBS Disturbance of Length 127	237
C-15	Bottom Composition Response for PRBS Disturbance of Length 127 using Steam	238
C-16	Bottom Composition Response for PRBS Disturbance of Length 127 using Reflux	239
C-17	Top Composition Response for PRBS Disturbance of Length 127 using Steam	240
C-18	Top Composition Response for PRBS Disturbance of Length 127 using Reflux	241

CHAPTER ONE

INTRODUCTION

Due to the rapid growth of process control in modern industry, there has been a greater effort to increase the ease and the accuracy of model identification. Since a process model is usually required before most of the modern control schemes can be applied, process identification is very important to control strategies and small errors in the model can lead to large control problems. Therefore good model estimates from noisy data is a must in modern industry.

This thesis deals with the theory, the structure and the application of a group of identification programs known as the UNiversity of ALberta Identification Program (UNALIP). Originally it was intended to augment UNALIP with a time series identification routine using the work of Kogekar [1] but the existence of a well documented APL time series package [18] on the University of Alberta's MTS system made the inclusion of Kogekar's work redundant.

The University of Alberta Identification Program is a modification of the Oxford Identification Package (OXIP). OXIP, developed for implementation on an IBM

1130 digital computer with 8K core memory capabilities, was obtained from Dr. D. W. Clarke of the Department of Engineering Science, University of Oxford, Oxford, England. UNALIP contains four identification algorithms along with various statistical and graphical options. The four identification methods are:

- 1) Generalized Least Squares (STIME)
- 2) Maximum Likelihood (ASTRM)
- 3) Instrumental Variables (IVOFF)
- 4) Unbiased Estimator (UBIAS)

The University of Alberta Identification Program attempts to identify the parameters of the pulse transfer function of the process under consideration. The process is treated as a 'black box'. Therefore no physical or mathematical description of the system is required, only the input/output data sequences are used. The source code is written in FORTRAN and the University of Alberta's Amdahl 470V/6 digital computer was used to implement the program.

Chapter 2 is an outline of the theory of the four identification methods. The nomenclature and symbols of the methods were intentionally kept consistent amongst themselves to avoid confusion.

A description of each of the major program modules was done in Chapter 3. An extensive overhaul of the program structure was required because the Amdahl computer's speed and size did not put limitations on the

data handling of the program as was the case originally when OXIP was implemented on an IBM 1130 digital computer. Matrices and vectors were substituted for file handling, the maximum model order was increased, the mechanical switches which controlled program flow were replaced by software switches, improved program commenting was accomplished, simpler coding and a switch to FORTRAN IV were a few of the changes that were instituted, resulting in UNALIP.

In Chapter 4, testing of the estimation routines was carried out. The procedure used to determine the accuracy and speed of the various routines was the same as that used by Isermann [2]. Three different processes were tested using generated data of three different pseudo random binary sequence run lengths and two different noise/signal ratios.

Chapter 5 contains the work done to estimate the pulse transfer functions for two pilot plant units. One unit was a liquid-liquid, concentric tube heat exchanger. The other unit was a binary distillation column with eight bubble cap trays.

CHAPTER TWO

THEORY OF IDENTIFICATION METHODS

2.1 INTRODUCTION

This section describes the theory of the four off-line identification algorithms used in the University of Alberta Identification Package. Most of the content of this chapter is not original but is taken from different sources that deal with the techniques under consideration. The nomenclature and the symbols have been kept consistent for all techniques and special attention has been directed to clarifying certain aspects of the theory not considered in detail in the original sources.

2.2 STATEMENT OF THE PROBLEM

Consider a discrete time, single input, single output system that can be described by the following linear, time invariant, difference equation.

$$v_t + \sum_{i=1}^n a_i v_{t-i} = \sum_{i=0}^n b_i u_{t-i-k} \quad (2.1)$$

- where: 1) v_t is the noise free output of the system
 2) u_t is the input of the system
 3) n is the order of the system
 4) k is the time delay of the system in terms of an integer number of sampling intervals
 5) a_i, b_i are the system parameters

Let the noise free output be disturbed by an independent zero mean additive noise signal.

$$y_t = v_t + \epsilon_t \quad (2.2)$$

- where: 1) y_t is the corrupted output of the system
 2) ϵ_t is the independent zero mean additive noise signal

The additive noise signal is considered to be the random process output from a general filter driven by a white noise source. This can be expressed as

$$\epsilon_t + \sum_{i=1}^{nn} c_i \epsilon_{t-i} = w_t + \sum_{i=1}^{nn} d_i w_{t-i} \quad (2.3)$$

where: 1) w_t is the white noise source of the general filter

2) nn is the order of the noise process (filter)

3) c_i, d_i are the noise process parameters

Equations (2.1) and (2.3) can be rewritten as

$$[1+A(z^{-1})]v_t = z^{-k}B(z^{-1})u_t \quad (2.4)$$

$$[1+C(z^{-1})]\epsilon_t = [1+D(z^{-1})]w_t \quad (2.5)$$

where: 1) $A(z^{-1}) = a_1 z^{-1} + \dots + a_n z^{-n}$

2) $B(z^{-1}) = b_0 + b_1 z^{-1} + \dots + b_n z^{-n}$

3) $C(z^{-1}) = c_1 z^{-1} + \dots + c_{nn} z^{-nn}$

4) $D(z^{-1}) = d_1 z^{-1} + \dots + d_{nn} z^{-nn}$

5) z is the z -transform variable used as the backward shift operator ($z^{-k}y_t = y_{t-k}$)

Multiplying both sides of equation (2.2) by $[1+A(z^{-1})]$ results in

$$[1+A(z^{-1})]y_t = [1+A(z^{-1})]v_t + [1+A(z^{-1})]\epsilon_t \quad (2.6)$$

which can be rewritten as

$$[1+A(z^{-1})]y_t = z^{-k}B(z^{-1})u_t + e_t \quad (2.7)$$

where $e_t = w_t [1+A(z^{-1})][1+D(z^{-1})]/[1+C(z^{-1})]$.

Equation (2.7) can be rearranged as

$$y_t = -A(z^{-1})y_t + z^{-k}B(z^{-1})u_t + e_t \quad (2.8)$$

To write equation (2.8) in standard vector matrix notation of regression analysis, consider a sequence of observations (u,y) to have been made of the system's input/output variables. Then assuming that $b_0=0$ for the system being considered

$$y_t = -a_1 y_{t-1} - \dots - a_n y_{t-n} + b_1 u_{t-1-k} + \dots + b_n u_{t-n-k} + e_t \quad (2.9)$$

$$y_{t+N} = -a_1 y_{t+N-1} - \dots - a_n y_{t+N-n} + b_1 u_{t+N-1-k} + \dots + b_n u_{t+N-n-k} + e_{t+N}$$

where N is the number of data pairs in the sequence (u,y) . Equation (2.9) can be written as

$$\underline{y} = -A(z^{-1})\underline{y} + z^{-k}B(z^{-1})\underline{u} + \underline{e} \quad (2.10)$$

where: 1) $\underline{y} = [y_t, \dots, y_{t+N}]^T$
 2) $\underline{u} = [u_t, \dots, u_{t+N}]^T$
 3) $\underline{e} = [e_t, \dots, e_{t+N}]^T$

Equation (2.10) can be expressed as

$$\underline{y} = [-z^{-1}\underline{y}, \dots, -z^{-n}\underline{y}, z^{-k-1}\underline{u}, \dots, z^{-k-n}\underline{u}] \begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix} + \underline{e} \quad (2.11)$$

which can be written as

$$\underline{y} = \underline{X}\underline{\theta} + \underline{e} \quad (2.12)$$

where: 1) $\underline{a} = [a_1, \dots, a_n]^T$

2) $\underline{b} = [b_1, \dots, b_n]^T$

$$3) \quad \underline{X} = \begin{bmatrix} (-y_{t-1}) \dots (-y_{t-n}) & u_{t-1-k} \dots u_{t-n-k} \\ \cdot \\ \cdot \\ \cdot \\ (-y_{t+N-1}) \dots (-y_{t+N-n}) & u_{t+N-1-k} \dots u_{t+N-n-k} \end{bmatrix}$$

4) $\underline{\theta} = [\underline{a}, \underline{b}]^T$

2.3 ASSUMPTIONS

In the statement of the problem, the following assumptions [3] were made:

- 1) the functions $[1+A(z^{-1})][1+D(z^{-1})]/[1+C(z^{-1})]$ and $[1+A(z^{-1})]$ all have their zeros inside the unit circle
- 2) there are no factors common to all three polynomials $[1+A(z^{-1})]$, $B(z^{-1})$ and $[1+D(z^{-1})]$
- 3) for the type of system to be investigated, $b_0=0$ and $b_1 \neq 0$

The assumption that the functions $[1+A(z^{-1})]$ and $[1+A(z^{-1})][1+D(z^{-1})]/[1+C(z^{-1})]$ have all their zeros inside the unit circle implies that the system is asymptotically stable. The second assumption implies that every state of the system is controllable either from \underline{u} or \underline{w} . For simplicity it is assumed that $b_0=0$ and $b_1 \neq 0$ [4].

2.4 GENERALIZED LEAST SQUARES

Generalized least squares is a technique of parameter estimation proposed by Clarke [7]. This technique attempts to produce unbiased estimates from noisy data using least squares by filtering the input/output sequences until the bias is removed. In this section the least squares technique is introduced and then Clarke's modifications are described.

The least squares estimation [5] problem is the selection of an estimate $\hat{\underline{\theta}}$ of $\underline{\theta}$ such that the loss function defined as

$$\begin{aligned} V(\underline{\theta}) &= (\underline{e}^T \underline{T}^{-1} \underline{e})/2 \\ &= (\underline{y} - \underline{X}\underline{\theta})^T \underline{T}^{-1} (\underline{y} - \underline{X}\underline{\theta})/2 \end{aligned} \quad (2.13a)$$

is minimized.

- where: 1) $\hat{\underline{\theta}}$ is the estimate of the parameter vector $\underline{\theta}$
- 2) \underline{T}^{-1} is a weighting matrix, an indication of the degree of confidence placed in the individual measurements [29]
- 3) \underline{e} is the generalized error for the system under consideration and can be obtained from (2.12)
- 4) $V(\underline{\theta})$ is the loss function of classical least squares

The loss function is minimized by carrying out a partial differentiation of $V(\underline{\theta})$.

$$\left. \frac{\partial V(\underline{\theta})}{\partial \underline{\theta}} \right|_{\underline{\theta}=\hat{\underline{\theta}}} = \underline{\underline{X}}^T \underline{\underline{T}}^{-1} (\underline{\underline{y}} - \underline{\underline{X}} \hat{\underline{\theta}}) = 0 \quad (2.13b)$$

For the case of $\underline{\underline{T}}$ taken as the identity matrix [6], then the least squares estimate of $\underline{\theta}$ is

$$\begin{aligned} \underline{\underline{X}}^T (\underline{\underline{y}} - \underline{\underline{X}} \hat{\underline{\theta}}) &= 0 \\ \hat{\underline{\theta}} &= [\underline{\underline{X}}^T \underline{\underline{X}}]^{-1} \underline{\underline{X}}^T \underline{\underline{y}} \end{aligned} \quad (2.13c)$$

Multiplying both sides of equation (2.12) by $[\underline{\underline{X}}^T \underline{\underline{X}}]^{-1} \underline{\underline{X}}^T$ and using equation (2.13c) results in

$$\hat{\underline{\theta}} = \underline{\theta} + [\underline{\underline{X}}^T \underline{\underline{X}}]^{-1} \underline{\underline{X}}^T \underline{\underline{e}} \quad (2.14)$$

Classical regression theory assumes that the elements of $\underline{\underline{X}}$ are independent of the elements of $\underline{\underline{e}}$ [7]. Therefore, due to the formulation of the problem, biased estimates will result because the elements of $\underline{\underline{X}}$ are dependent on the elements of $\underline{\underline{e}}$ (cf. equations (2.2), (2.5) and (2.7)). In this instance, the bias remains regardless of the number of data pairs [7]. This can be shown by rearranging equation (2.14) and taking the expectation of both sides of the resulting equation

$$E([\underline{X}^T \underline{X}](\hat{\underline{\theta}} - \underline{\theta})) = E(\underline{X}^T \underline{e}) \quad (2.15)$$

From equation (2.7) and using a vector representation as in equation (2.10), \underline{e} can be expressed as

$$\underline{e} = \frac{[1+A(z^{-1})][1+D(z^{-1})]\underline{w}}{[1+C(z^{-1})]} \quad (2.16)$$

where $\underline{w} = [w_t, \dots, w_{t+N}]^T$.

Equation (2.16) can be rewritten as

$$\underline{e} = [1+F(z^{-1})]\underline{w} \quad (2.17)$$

where: 1) $F(z^{-1}) = f_1 z^{-1} + f_2 z^{-2} + \dots$

2) f_i are the coefficients of an infinite sequence produced by the long division expansion of the expression

$$[1+A(z^{-1})][1+D(z^{-1})]/[1+C(z^{-1})]$$

This allows the right hand side of equation (2.15) to be expressed as [7]:

$$\begin{aligned}
E(\underline{X}^T \underline{e}) &= E \left[\begin{bmatrix} -z^{-1} \underline{y}^T \\ \vdots \\ -z^{-n} \underline{y}^T \\ z^{-k-1} \underline{u}^T \\ \vdots \\ z^{-k-n} \underline{u}^T \end{bmatrix} [\underline{w} + f_1 z^{-1} \underline{w} + \dots] \right] \\
&= E \left[\begin{bmatrix} -z^{-1} \underline{y}^T (\underline{w} + f_1 z^{-1} \underline{w} + \dots) \\ \vdots \\ -z^{-n} \underline{y}^T (\underline{w} + f_1 z^{-1} \underline{w} + \dots) \\ z^{-k-1} \underline{u}^T (\underline{w} + f_1 z^{-1} \underline{w} + \dots) \\ \vdots \\ z^{-k-n} \underline{u}^T (\underline{w} + f_1 z^{-1} \underline{w} + \dots) \end{bmatrix} \right] \\
&= E \left[\begin{bmatrix} -\langle z^{-1} \underline{y}, \underline{w} \rangle - f_1 \langle z^{-1} \underline{y}, z^{-1} \underline{w} \rangle - \dots \\ \vdots \\ -\langle z^{-n} \underline{y}, \underline{w} \rangle - f_1 \langle z^{-n} \underline{y}, z^{-1} \underline{w} \rangle - \dots \\ \langle z^{-k-1} \underline{u}, \underline{w} \rangle + f_1 \langle z^{-k-1} \underline{u}, z^{-1} \underline{w} \rangle + \dots \\ \vdots \\ \langle z^{-k-n} \underline{u}, \underline{w} \rangle + f_1 \langle z^{-k-n} \underline{u}, z^{-1} \underline{w} \rangle + \dots \end{bmatrix} \right] \quad (2.18)
\end{aligned}$$

where $\langle \underline{l}, \underline{m} \rangle$ is the inner product of a pair of s dimensional vectors \underline{l} and \underline{m} with the inner product

$$\text{defined as } \langle \underline{l}, \underline{m} \rangle = \sum_{i=1}^s l_i m_i .$$

Substituting the vector form of equation (2.7) into the right hand side of equation (2.18) allows the expectation of the inner product of \underline{y} and \underline{w} to be expressed as

$$E(\langle \underline{y}, \underline{w} \rangle) = E \left[\left\langle \frac{z^{-k} B(z^{-1}) \underline{u}}{[1+A(z^{-1})]} + \frac{\underline{e}}{[1+A(z^{-1})]}, \underline{w} \right\rangle \right] \quad (2.19a)$$

and from equation (2.16) it follows that

$$E(\langle \underline{y}, \underline{w} \rangle) = E \left[\left\langle \frac{z^{-k} B(z^{-1}) \underline{u}}{[1+A(z^{-1})]}, \underline{w} \right\rangle + \left\langle \frac{[1+D(z^{-1})] \underline{w}}{[1+C(z^{-1})]}, \underline{w} \right\rangle \right] \quad (2.19b)$$

if it can be assumed that \underline{u} and \underline{w} are uncorrelated then [7]:

$$E(\langle z^{-p} \underline{u}, z^{-j} \underline{w} \rangle) = 0 \quad \text{for all } p \text{ and } j \quad (2.20)$$

so equation (2.19b) becomes

$$E(\langle \underline{y}, \underline{w} \rangle) = E \left[\left\langle \frac{[1+D(z^{-1})] \underline{w}}{[1+C(z^{-1})]}, \underline{w} \right\rangle \right] \quad (2.21)$$

From equations (2.7) and (2.17) it can be seen that y_t is independent of w_{t+1}, w_{t+2}, \dots , but dependent on w_t, w_{t-1}, \dots , therefore let

$$1+F^*(z^{-1}) = \frac{1+D(z^{-1})}{1+C(z^{-1})} \quad (2.22)$$

where: 1) $F^*(z^{-1}) = f_1^* z^{-1} + f_2^* z^{-2} + \dots$

2) f_j^* are the coefficients of the infinite sequence produced by the long division expansion of the expression
 $[1+D(z^{-1})]/[1+C(z^{-1})]$

The right hand side of equation (2.21) can be expressed as [7]:

$$\begin{aligned} E(\langle z^{-i}(1+F^*(z^{-1}))\underline{w}, z^{-j}\underline{w} \rangle) &= f_{j-i}^* \sigma^2 \quad \text{if } j > i \\ &= \sigma^2 \quad \text{if } j = i \\ &= 0 \quad \text{if } j < i \end{aligned} \quad (2.23)$$

where σ^2 is the variance of the white noise source.

Equation (2.18) becomes

$$E(\underline{X}^T \underline{e}) = E \left[\begin{array}{c} -f_1 \sigma^2 - f_2 f_1^* \sigma^2 - f_3 f_2^* \sigma^2 - \dots \\ \vdots \\ -f_n \sigma^2 - f_{n+1} f_1^* \sigma^2 - f_{n+2} f_2^* \sigma^2 - \dots \\ 0 + 0 + \dots \\ \vdots \\ 0 + 0 + \dots \end{array} \right] \quad (2.24)$$

The right hand side of equation (2.15) is non-zero unless the coefficients $f_i = 0$ for all i , which occurs [7] only when $[1+A(z^{-1})][1+D(z^{-1})] = [1+C(z^{-1})]$ which is not generally true. The estimates, $\hat{\underline{\theta}}$, remain asymptotically biased unless \underline{e} is an uncorrelated sequence. Note that even if ϵ_t in equation (2.2) is a white noise sequence, the estimates of $\underline{\theta}$ are still biased because e_t is not a white noise sequence [7].

If the coefficients of $A(z^{-1})$, $C(z^{-1})$ and $D(z^{-1})$ were known, then it would be possible to modify \underline{y} and \underline{u} by a moving average filter with coefficients of the autoregressive model of the noise process to give unbiased estimates of $\underline{\theta}$ [8]. The filtering process can be represented as

$$\underline{u}^* = [1 + R_0(z^{-1})]\underline{u} \quad (2.25)$$

$$\underline{y}^* = [1 + R_0(z^{-1})]\underline{y} \quad (2.26)$$

- where: 1) \underline{u}^* is the filtered input of the system
- 2) \underline{y}^* is the filtered corrupted output of the system
- 3) $R_o(z^{-1}) = r_{o_1}z^{-1} + \dots + r_{o_{IRO}}z^{-IRO}$
- 4) r_{o_i} are the coefficients of the autoregressive model of the noise process
- 5) IRO is the order of the autoregressive model of the noise process

If IRO is infinite, the coefficients of the moving average filter are given by [8]:

$$[1+R_o(z^{-1})] = [1+C(z^{-1})]/\{[1+A(z^{-1})][1+D(z^{-1})]\} \quad (2.27)$$

then from equations (2.2), (2.5), (2.25), (2.26) and (2.27) the following equation can be obtained [7]:

$$\begin{aligned} \underline{y}^* &= \underline{v}^* + \frac{[1+C(z^{-1})]\underline{\varepsilon}}{[1+A(z^{-1})][1+D(z^{-1})]} \\ &= \underline{v}^* + \frac{\underline{w}}{[1+A(z^{-1})]} \end{aligned} \quad (2.28)$$

Equation (2.28) can be rewritten using equation (2.4) to give

$$[1+A(z^{-1})]\underline{y}^* = z^{-k}B(z^{-1})\underline{u}^* + \underline{w} \quad (2.29)$$

Since \underline{w} is uncorrelated white noise, a least squares estimate using the filtered input/output is unbiased. The new least squares is

$$\underline{\hat{\theta}}^* = [\underline{X}^{*T}\underline{X}^*]^{-1}\underline{X}^{*T}\underline{y}^* \quad (2.30)$$

where: 1) \underline{X}^* is the filtered result of the matrix \underline{X}
 2) $\underline{\hat{\theta}}^*$ is the estimate of the parameter vector $\underline{\theta}$
 for the filtered data

Now the problem is one of finding a suitable moving average filter so that the input/output data can be modified to allow unbiased estimates to be obtained by a least squares estimator from this modified input/output data. This problem would be trivial if the coefficients of $R_o(z^{-1})$ were available but the polynomial coefficients of $A(z^{-1})$, $C(z^{-1})$ and $D(z^{-1})$ are not known. Clarke [7] solves this problem by designing a filter recursively until the final estimates are unbiased. It is assumed [8] that the estimates of the model error are a good representation of the noise sequence \underline{e} .

$$\underline{s}_t = \underline{y}_t - \tilde{\underline{y}}_t \quad (2.31)$$

where: 1) \underline{s}_t is the estimate of the model error
 (residuals)

2) \tilde{y}_t is the output from the calculated model of the estimation routine

Using equation (2.27) and the definition of \underline{e} , \underline{w} can be expressed as

$$\underline{w} = [1 + Ro(z^{-1})]\underline{e} \quad (2.32)$$

then using the assumption $\underline{e} \cong \underline{s}$, equation (2.32) becomes

$$\underline{w} \cong [1 + Ro(z^{-1})]\underline{s} \quad (2.33)$$

which can be expressed as

$$\underline{s} = \underline{G}\underline{h} + \underline{w} \quad (2.34)$$

where: 1) $\underline{G} = [-z^{-1}\underline{s}, \dots, -z^{-IRO}\underline{s}]$

2) $\underline{h} = [ro_1, \dots, ro_{IRO}]^T$

The least squares estimate of \underline{h} is

$$\hat{\underline{h}} = [\underline{G}^T \underline{G}]^{-1} \underline{G}^T \underline{s} \quad (2.35)$$

where $\hat{\underline{h}}$ is the estimate of the parameter vector \underline{h} . This estimate is unbiased because the elements of \underline{G} are independent of \underline{w} . The estimation of the autoregressive noise process model parameters allows the filtering of the input/output data to be performed. Thus in the limit as $\hat{\underline{h}}$ approaches \underline{h} , the estimates $\hat{\underline{\theta}}$ will be

unbiased [9]. Since a finite approximation is used for the autoregressive noise process model parameters and \underline{s} was used for \underline{e} , $\hat{\underline{\theta}}^*$ is still biased and iterations must be employed to remove the bias. For each iteration, new residuals \underline{s}^* are evaluated using \underline{u} , \underline{y} and $\hat{\underline{\theta}}^*$ from the previous iteration using the relation

$$\underline{s}^* = \underline{y} - \underline{X}^T \hat{\underline{\theta}}^* \quad (2.36)$$

where \underline{s}^* is the new estimate of the model error.

The generalized least squares procedure is a "quasilinear formulation of a non-linear estimation problem" [8]. Convergence is not guaranteed unless restrictive assumptions are made about the closeness of the original parameter guesses to the actual values [8].

2.5 MAXIMUM LIKELIHOOD

The maximum likelihood estimation technique can be derived from the Bayes' estimator [9]. There are three assumptions used to derive the Bayes' estimator:

- a) the probability density functions of the parameters a_i , b_i and f_i are known
- b) the loss function which determines the cost of choosing the estimates of a_i , b_i and f_i is known
- c) the probability density function of the noise sequence is known

The maximum likelihood estimator drops assumptions (a) and (b), thus the parameters a_i , b_i and f_i are no longer random variables but known constant parameters. Astrom et al [3] have used the maximum likelihood estimator to identify process parameters of systems that can be characterized by linear, time invariant, difference equations. This section deals with the maximum likelihood technique as employed by Astrom et al [4].

The linear, time invariant, difference equation can be obtained by using the vector forms of equations (2.5), (2.7), and (2.17) to give

$$[1+F(z^{-1})]\underline{w} = [1+A(z^{-1})]\underline{y} - z^{-k}B(z^{-1})\underline{u} \quad (2.37)$$

A priori, a sample of the corrupted output sequence

will be random variables that have a joint conditional probability density function represented as

$$p[\underline{y}|\underline{u},\underline{a},\underline{b},\underline{f}] \quad (2.38)$$

where: 1) $p[\underline{y}|\underline{u},\underline{a},\underline{b},\underline{f}]$ is the joint conditional probability density function of the output given the input, the system parameters and the noise process parameters

$$2) \underline{f} = [f_1, \dots]^T$$

A posteriori, the observed values of the random variable are known from direct measurement.

$$y_t = y'_t, \dots, y_{t+N} = y'_{t+N}$$

The problem is thus one of determining $\hat{\underline{a}}$, $\hat{\underline{b}}$, and $\hat{\underline{f}}$ from \underline{y}' .

where: 1) $\hat{\underline{a}}$ is the estimate of the parameter vector \underline{a}

2) $\hat{\underline{b}}$ is the estimate of the parameter vector \underline{b}

3) $\hat{\underline{f}}$ is the estimate of the parameter vector \underline{f}

$$4) \underline{y}' = [y'_t, \dots, y'_{t+N}]^T$$

5) y'_t is the observed corrupted output of the system.

The functional relationship between \underline{y}' and $\hat{\underline{a}}$, $\hat{\underline{b}}$ and $\hat{\underline{f}}$ is the same as the functional relationship between \underline{y} and \underline{a} , \underline{b} and \underline{f} [3]. This relationship is written as

$$L[\underline{y}'|\underline{u},\hat{\underline{a}},\hat{\underline{b}},\hat{\underline{f}}] \quad (2.39)$$

and is called the likelihood function. Usually $\ln(L[y'|\underline{u},\hat{\underline{a}},\hat{\underline{b}},\hat{\underline{f}}])$ is used [4] because the logarithmic function is monotonic and the maxima and minima of $L[y'|\underline{u},\hat{\underline{a}},\hat{\underline{b}},\hat{\underline{f}}]$ occur at the same values as the maxima and minima of $p[y|\underline{u},\underline{a},\underline{b},\underline{f}]$. The logarithm of the likelihood function has the useful property [3] of continuous partial derivatives for all parameters. Since the minima are finite, the gradient of the function will thus vanish at the minima. Therefore values of $\hat{\underline{a}}$, $\hat{\underline{b}}$ and $\hat{\underline{f}}$ can be obtained by solving

$$\left. \frac{\partial}{\partial \underline{\varnothing}} \ln L[y'|\underline{u},\underline{\varnothing}] \right|_{\underline{\varnothing} = \hat{\underline{\varnothing}}} = 0 \quad (2.40)$$

where: 1) $\underline{\varnothing} = [\underline{a},\underline{b},\underline{f}]^T$

2) $\hat{\underline{\varnothing}} = [\hat{\underline{a}},\hat{\underline{b}},\hat{\underline{f}}]^T$

The solution must also satisfy

$$\left. \frac{\partial^2}{\partial \underline{\varnothing}^2} \ln L[y'|\underline{u},\underline{\varnothing}] \right|_{\underline{\varnothing} = \hat{\underline{\varnothing}}} < 0 \quad (2.41)$$

Based on the assumption that the noise sequence \underline{w} has a Gaussian distribution [4], the likelihood function of \underline{w} is

$$L[\underline{w}] = \frac{1}{(2\pi)^{N/2} \sigma^N} \left(\exp \frac{-\underline{w}^T \underline{w}}{2\sigma^2} \right) \quad (2.42)$$

where σ is the standard deviation of the noise sequence. Taking the logarithm of both sides of equation (2.42) results in

$$\ln L[\underline{w}] = -\frac{N \ln 2\pi}{2} - N \ln \sigma - \frac{\underline{w}^T \underline{w}}{2\sigma^2} \quad (2.43)$$

The joint probability density function of \underline{y} and $\underline{\gamma}$ is defined as [10]

$$p(\underline{y}, \underline{\gamma}) = p(\underline{\gamma}, \underline{w}) \left| \frac{\partial \underline{w}}{\partial \underline{y}} \right| \quad (2.44)$$

where: 1) $\underline{\gamma} = [\underline{u}, \hat{\underline{a}}, \hat{\underline{b}}, \hat{\underline{f}}]$

2) $|\partial \underline{w} / \partial \underline{y}|$ is the absolute value of the Jacobian of \underline{y}

Since \underline{y} is a one to one transformation of \underline{w} and the determinant of the identity matrix is unity, equation (2.44) becomes

$$p(\underline{y}, \underline{\gamma}) = p(\underline{\gamma}, \underline{w}) \quad (2.45)$$

and combined with the definition of conditional probability [11], the following equation results:

$$\begin{aligned}
p(\underline{y}|\underline{\gamma}) &= \frac{p(\underline{y}, \underline{\gamma})}{p(\underline{\gamma})} \\
&= \frac{p(\underline{\gamma}, \underline{w})}{p(\underline{\gamma})}
\end{aligned}
\tag{2.46}$$

If $\underline{\gamma}$ and \underline{w} are independent then [11]:

$$p(\underline{\gamma}, \underline{w}) = p(\underline{\gamma})p(\underline{w}) \tag{2.47}$$

and using equations (2.46) and (2.47) gives the following expression

$$p(\underline{y}|\underline{\gamma}) = \frac{p(\underline{\gamma})p(\underline{w})}{p(\underline{\gamma})} = p(\underline{w}) \tag{2.48}$$

Now returning to the likelihood function, it follows that

$$\begin{aligned}
\ln L[\underline{y}'|\underline{\gamma}] &= \ln L[\underline{w}] \\
&= -\frac{N \ln 2\pi}{2} - N \ln \sigma - \frac{\underline{w}^T \underline{w}}{2\sigma^2}
\end{aligned}
\tag{2.49}$$

The likelihood function is thus a function of the parameters $\hat{\underline{a}}$, $\hat{\underline{b}}$, $\hat{\underline{f}}$ and σ as well as the n initial conditions for equation (2.36). For simplicity it is assumed that the initial conditions are zero [4].

As in generalized least squares, a loss function is introduced which gives a measure of parameter estimation accuracy. The loss function that is used can be expressed as

$$V(\underline{\theta}) = (\underline{w}^T \underline{w})/2 \quad (2.50)$$

It can be seen that minimizing the loss function is equivalent to maximizing $\ln([\underline{y}|\underline{y}])$ because equation (2.49) has its maximum value when $(\underline{w}^T \underline{w})/2$ is a minimum since the other two terms of the right hand side of equation (2.49) are constant and do not affect the maximization of $\ln([\underline{y}|\underline{y}])$. The variance, σ^2 , of the noise sequence can be determined when $\hat{\underline{\theta}}$ is known such that $V(\hat{\underline{\theta}})$ is a minimum by using the relationship

$$\sigma^2 = \min(\underline{w}^T \underline{w})/N \quad (2.51)$$

where $\min(\underline{w}^T \underline{w})$ is the minimum value of $\underline{w}^T \underline{w}$.

The goal is to obtain $\hat{\underline{a}}$, $\hat{\underline{b}}$ and $\hat{\underline{f}}$ such that $V(\hat{\underline{\theta}})$ is a minimum. This could be done analytically if the loss function had a simple dependence on $\hat{\underline{a}}$, $\hat{\underline{b}}$ and $\hat{\underline{f}}$ but examination of equations (2.36) and (2.50) leads to the conclusion that $V(\underline{\theta})$ is a quadratic function of \underline{a} and \underline{b} while the dependence of \underline{f} is more complex [4]. Therefore an analytical solution cannot be obtained and an iterative approach must be sought. Due to the

structure of the problem, a combined gradient Gauss-Newton technique where fast convergence is obtained through the use of second partial derivatives was originally employed by Astrom et al [3]. However, the method used in UNALIP is the modified Gauss-Newton-Marquardt algorithm [9] using a modified Levenburg parameter [12]. An update of the parameter vector using this method is calculated by the following equation

$$\hat{\underline{\theta}}^{t+1} = \hat{\underline{\theta}}^t - [\psi \hat{\underline{S}}_{\hat{\underline{\theta}}\hat{\underline{\theta}}}(\hat{\underline{\theta}}^t)]^{-1} \hat{\underline{S}}_{\hat{\underline{\theta}}}(\hat{\underline{\theta}}^t) \quad (2.52)$$

where: 1) ψ is the modified Levenburg parameter

2) $\hat{\underline{S}}_{\hat{\underline{\theta}}\hat{\underline{\theta}}}$ is the matrix of second partial derivatives of the loss function

3) $\hat{\underline{S}}_{\hat{\underline{\theta}}}$ is the gradient of the loss function

4) $\hat{\underline{\theta}}^t$ is the value of the estimated parameters at the "t"th iteration

Since \underline{w} is not available, an approximation is made to determine the noise source using the following relationship obtained from equation (2.36)

$$\hat{\underline{w}} = \{[1+A(z^{-1})]\underline{y} - z^{-k}B(z^{-1})\underline{u}\} / [1+F(z^{-1})] \quad (2.53)$$

where: 1) $\hat{\underline{w}}$ is the approximated noise source of the model

2) $1/[1+F(z^{-1})]$ is an autoregressive filter

Thus the loss function becomes

$$V(\underline{\hat{\theta}}) = \underline{\hat{w}}^T \underline{\hat{w}} / 2 = \frac{1}{2} \sum_{t=1}^N \hat{w}_t^2 \quad (2.54)$$

where \hat{w}_t is an element of the noise sequence $\underline{\hat{w}}$ and the maximum likelihood estimate of the variance of the noise sequence becomes

$$\hat{\sigma}^2 = \min(\underline{\hat{w}}^T \underline{\hat{w}}) / N \quad (2.55)$$

where $\hat{\sigma}^2$ is the estimate of the variance of the noise sequence.

The partial derivatives of the loss function are obtained by differentiating equation (2.54) to give

$$\underline{S}_{\underline{\hat{\theta}}}^{\hat{\theta}} = \frac{\partial V(\underline{\hat{\theta}})}{\partial \hat{\theta}_i} = \sum_{t=1}^N \hat{w}_t \frac{\partial \hat{w}_t}{\partial \hat{\theta}_i} \quad (2.56)$$

$$\underline{S}_{\underline{\hat{\theta}}\underline{\hat{\theta}}}^{\hat{\theta}\hat{\theta}} = \frac{\partial^2 V(\underline{\hat{\theta}})}{\partial \hat{\theta}_i \partial \hat{\theta}_j} = \sum_{t=1}^N \frac{\partial \hat{w}_t}{\partial \hat{\theta}_i} \frac{\partial \hat{w}_t}{\partial \hat{\theta}_j} + \sum_{t=1}^N \hat{w}_t \frac{\partial^2 \hat{w}_t}{\partial \hat{\theta}_i \partial \hat{\theta}_j} \quad (2.57)$$

where $\hat{\theta}_i$ is the "i"th element of the parameter vector $\underline{\hat{\theta}}$. The derivatives of \hat{w}_t , used in equations (2.56) and

(2.57), are obtained by differentiating equation (2.53) as follows

$$\frac{\partial \hat{w}_t}{\partial \hat{a}_i} = z^{-i} y_t / [1 + F(z^{-1})] \quad (2.58)$$

$$\frac{\partial \hat{w}_t}{\partial \hat{b}_i} = -z^{-i-k} u_t / [1 + F(z^{-1})] \quad (2.59)$$

$$\frac{\partial \hat{w}_t}{\partial \hat{f}_i} = -z^{-i} \hat{w}_t / [1 + F(z^{-1})] \quad (2.60)$$

$$\frac{\partial^2 \hat{w}_t}{\partial \hat{a}_i \partial \hat{f}_j} = -z^{-i-j} \frac{\partial \hat{w}_t}{\partial \hat{a}_i} \quad (2.61)$$

$$\frac{\partial^2 \hat{w}_t}{\partial \hat{b}_i \partial \hat{f}_j} = -z^{-i-j} \frac{\partial \hat{w}_t}{\partial \hat{b}_i} \quad (2.62)$$

$$\frac{\partial^2 \hat{w}_t}{\partial \hat{f}_i \partial \hat{f}_j} = -2z^{-i-j} \frac{\partial \hat{w}_t}{\partial \hat{f}_i} \quad (2.63)$$

where \hat{a}_i , \hat{b}_i , and \hat{f}_i are the "i"th elements of the vectors \hat{a} , \hat{b} and \hat{f} . Notice that the second order partial derivatives of \hat{w} are all identically equal to zero [4]. Thus, equation (2.52) can be used to solve for $\hat{\theta}^{t+1}$.

2.6 INSTRUMENTAL VARIABLES

A simple solution to the problem of bias in parameter estimates due to observations contaminated with noise is the instrumental variable technique. The instrumental variable method is the same as the least squares technique except that the bias is eliminated by filtering equation (2.12) using the instrumental variable matrix. This section deals with the instrumental variable technique as it is used in UNALIP. Using equation (2.12), the generalized error for the system under consideration can be written as

$$\underline{e} = \underline{y} - \underline{X}\underline{\theta} \quad (2.64)$$

Unbiased estimates of $\underline{\theta}$ can be obtained from a least squares solution if both sides of equation (2.64) are premultiplied by \underline{W}^T [13]:

$$\underline{W}^T \underline{e} = \underline{W}^T \underline{y} - \underline{W}^T \underline{X} \underline{\theta} \quad (2.65)$$

where \underline{W} is the instrumental variable matrix which must satisfy the following conditions:

- a) $E[\underline{W}^T \underline{e}] = 0$
- b) $\underline{W}^T \underline{X}$ is nonsingular

If the elements of the instrumental variable matrix are

chosen to be uncorrelated with the residuals then $E[\underline{W}^T \underline{e}] = 0$ and the problem reduces to a least squares analysis with no bias [14]. Using a similar loss function as that described in section 2.4

$$V(\underline{\theta}) = (\underline{W}^T \underline{e})^T \underline{W}^T \underline{e} / 2 \quad (2.66a)$$

and applying the least squares method to this loss function results in the following equations

$$V(\underline{\theta}) = (\underline{W}^T \underline{y} - \underline{W}^T \underline{X} \underline{\theta})^T (\underline{W}^T \underline{y} - \underline{W}^T \underline{X} \underline{\theta}) / 2$$

$$\left. \frac{\partial V(\underline{\theta})}{\partial \underline{\theta}} \right|_{\underline{\theta} = \hat{\underline{\theta}}} = \underline{W}^T \underline{X}^T (\underline{W} \underline{y} - \underline{W}^T \underline{X} \hat{\underline{\theta}}) = 0$$

$$\underline{W}^T \underline{y} = \underline{W}^T \underline{X} \hat{\underline{\theta}}$$

$$\hat{\underline{\theta}} = [\underline{W}^T \underline{X}]^{-1} \underline{W}^T \underline{y} \quad (2.66b)$$

A problem is encountered in trying to find the elements of the instrumental variable matrix that will satisfy the stated conditions [14]. It has been shown that there exists optimal instrumental variables [15]. In UNALIP, the input sequence, \underline{u} , and the calculated undisturbed output sequence $\tilde{\underline{y}}$ are used to construct the instrumental variable matrix. The calculated undisturbed output sequence is obtained by taking the

parameter estimates to be the parameters of the calculated model. The instrumental variable matrix becomes

$$\underline{W} = \begin{bmatrix} (-\tilde{y}_{t-1}) \dots (-\tilde{y}_{t-n}) u_{t-1-k} \dots u_{t-n-k} \\ \cdot \\ \cdot \\ \cdot \\ (-\tilde{y}_{t+N-1}) \dots (-\tilde{y}_{t+N-n}) u_{t+N-1-k} \dots u_{t+N-n-k} \end{bmatrix} \quad (2.67)$$

2.7 UNBIASED ESTIMATOR

The problem of bias in parameter estimates due to observations contaminated with noise is dealt with by introducing a new loss function to eliminate the bias. James et al [16] first show why the conventional least squares loss function can not handle the type of system being dealt with and then set forth a loss function which will handle the system.

Using the generalized error, equation (2.64), the loss function can be defined as [16]:

$$V(\underline{\theta}) = \underline{e}^T \underline{e} = (\underline{y} - \underline{X}\underline{\theta})^T (\underline{y} - \underline{X}\underline{\theta}) \quad (2.68)$$

Conventional linear least squares estimates are obtained by minimization of this loss function.

$$\left. \frac{\partial V(\underline{\theta})}{\partial \underline{\theta}} \right|_{\underline{\theta} = \hat{\underline{\theta}}} = 0 \quad (2.69)$$

The authors [16] of this unbiased estimator method manipulate the loss function to show that the classic loss function is of no use for the type of system considered. This is done by taking the expectation of both sides of equation (2.68) to give

$$E(V(\underline{\theta})) = E((\underline{y} - \underline{X}\underline{\theta})^T(\underline{y} - \underline{X}\underline{\theta})) \quad (2.70)$$

Equation (2.70) can be broken up into simpler parts by using equation (2.2) and the definition of \underline{X}

$$E(V(\underline{\theta})) = E((\underline{v} + \underline{\varepsilon} - (\underline{P} + \underline{S})\underline{\theta})^T(\underline{v} + \underline{\varepsilon} - (\underline{P} + \underline{S})\underline{\theta})) \quad (2.71)$$

where: 1)

$$\underline{P} = \begin{bmatrix} (-v_{t-1}) \dots (-v_{t-n}) u_{t-1-k} \dots u_{t-n-k} \\ \vdots \\ (-v_{t+N-1}) \dots (-v_{t+N-n}) u_{t+N-1-k} \dots u_{t+N-n-k} \end{bmatrix}$$

2)

$$\underline{S} = \begin{bmatrix} (-\varepsilon_{t-1}) \dots (-\varepsilon_{t-n}) 0 \dots 0 \\ \vdots \\ (-\varepsilon_{t+N-1}) \dots (-\varepsilon_{t+N-n}) 0 \dots 0 \end{bmatrix}$$

Equation (2.71) can be manipulated in the following way

$$\begin{aligned}
E(V(\underline{\theta})) &= E((\underline{v} + \underline{\varepsilon} - (\underline{P} + \underline{S})\underline{\theta})^T (\underline{v} + \underline{\varepsilon} - (\underline{P} + \underline{S})\underline{\theta})) \\
&= E((\underline{v} - \underline{P}\underline{\theta})^T (\underline{v} - \underline{P}\underline{\theta}) + 2(\underline{v} - \underline{P}\underline{\theta})^T (\underline{\varepsilon} - \underline{S}\underline{\theta}) \\
&\quad + (\underline{\varepsilon} - \underline{S}\underline{\theta})^T (\underline{\varepsilon} - \underline{S}\underline{\theta})) \\
&= E((\underline{v} - \underline{P}\underline{\theta})^T (\underline{v} - \underline{P}\underline{\theta})) + 2E((\underline{v} - \underline{P}\underline{\theta})^T (\underline{\varepsilon} - \underline{S}\underline{\theta})) \\
&\quad + E((\underline{\varepsilon} - \underline{S}\underline{\theta})^T (\underline{\varepsilon} - \underline{S}\underline{\theta})) \\
&= (\underline{v} - \underline{P}\underline{\theta})^T (\underline{v} - \underline{P}\underline{\theta}) + 2[E(\underline{v}^T \underline{\varepsilon}) - \underline{\theta}^T E(\underline{S}^T \underline{v}) \\
&\quad - \underline{\theta}^T E(\underline{P}^T \underline{\varepsilon}) + \underline{\theta}^T E(\underline{P}^T \underline{S}) \underline{\theta}] + E(\underline{\varepsilon}^T \underline{\varepsilon}) \\
&\quad - 2\underline{\theta}^T E(\underline{S}^T \underline{\varepsilon}) + \underline{\theta}^T E(\underline{S}^T \underline{S}) \underline{\theta} \quad (2.72)
\end{aligned}$$

and remembering that the expectation of two uncorrelated sequences is zero [17], the following equations can be written

$$\begin{aligned}
E(\underline{v}^T \underline{\varepsilon}) &= \underline{0} \\
E(\underline{P}^T \underline{v}) &= \underline{0} \\
E(\underline{P}^T \underline{\varepsilon}) &= \underline{0} \\
E(\underline{P}^T \underline{S}) &= \underline{0}
\end{aligned}$$

where: 1) $\underline{0}$ is the null vector
 2) $\underline{0}$ is the null matrix

So equation (2.72) can be rewritten as

$$E(V(\underline{\theta})) = (\underline{v} - \underline{P}\underline{\theta})^T (\underline{v} - \underline{P}\underline{\theta}) + \underline{\theta}^T E(\underline{S}^T \underline{S}) \underline{\theta} - \underline{\theta}^T E(\underline{S}^T \underline{\varepsilon}) + E(\underline{\varepsilon}^T \underline{\varepsilon}) \quad (2.73)$$

The first term of equation (2.73) is the conventional linear least squares loss function used when noise free data is available. The second and third terms lead to the bias in the estimates while the last term is related to the variance of the estimates [16].

The method of James et al [16] is to reject the conventional loss function, equation (2.68), and use the following loss function to eliminate the effects of the second and third terms of equation (2.73).

$$J(\underline{\theta}) = (\underline{y} - \underline{X}\underline{\theta})^T (\underline{y} - \underline{X}\underline{\theta}) - \underline{\theta}^T \underline{R}\underline{\theta} + 2\underline{\theta}^T \underline{Q} \quad (2.74)$$

where: 1) $\underline{R} = \begin{bmatrix} \underline{R}_c & | & \underline{0} \\ \hline \underline{0} & | & \underline{0} \end{bmatrix}$

2) $\underline{R}_c = [r_{ij}] = E(z^{-i} \underline{\varepsilon}^T z^{-j} \underline{\varepsilon}) \quad (i, j = 1, \dots, n)$

3) $\underline{Q} = [\underline{Q}_c \quad \underline{0}]$

4) $\underline{Q}_c = [q_i] = E(\underline{\varepsilon}^T z^{-i} \underline{\varepsilon}) \quad (i = 1, \dots, n)$

The estimate of the parameter vector $\underline{\theta}$ is then determined by classical least squares [5] to give:

$$\frac{\partial J(\underline{\theta})}{\partial \underline{\theta}} \bigg|_{\underline{\theta} = \hat{\underline{\theta}}} = 2\underline{Q} - 2\underline{R}\hat{\underline{\theta}} - 2\underline{X}^T(\underline{y} - \underline{X}\hat{\underline{\theta}}) = 0$$

$$\underline{X}^T \underline{y} - \underline{Q} = (\underline{X}^T \underline{X} - \underline{R}) \hat{\underline{\theta}}$$

$$\hat{\underline{\theta}} = (\underline{X}^T \underline{X} - \underline{R})^{-1} (\underline{X}^T \underline{y} - \underline{Q}) \quad (2.75)$$

CHAPTER THREE

ORGANIZATION AND STRUCTURE OF THE UNIVERSITY OF ALBERTA IDENTIFICATION PACKAGE

3.1 INTRODUCTION

This chapter details the purpose, method, flow diagrams, box diagrams, variable descriptions and subroutines called by the major program modules of the University of Alberta Identification Package. In general, the package accepts input/output data from noisy, single input/output systems, and estimates the best discrete, pulse transfer function models for the system and the noise processes. It can handle systems with time delays, integrations and nonstationary noise processes. The model obtained is of the form

$$\underline{y} = \frac{z^{-k} B(z^{-1}) \underline{u}}{[1+A(z^{-1})](1-z^{-1})^{INT}} + \frac{[1+D(z^{-1})] \underline{w}}{[1+C(z^{-1})](1-z^{-1})^{IUDIF}} \quad (3.1)$$

where: 1) $INT = IYDIF - IUDIF$

2) IUDIF is the number of differencings of the input sequence

3) IYDIF is the number of differencings of the output sequence

The optimal model must pass the following tests:

1) mean square error improvement over a simpler model

2) significance of individual parameters

3) nonsingularity of the parameter covariance matrix

4) uncorrelated residual sequences

5) zero cross correlations between the input signal and the error signal

There are certain restrictions placed on various variables due to matrix or vector size and logical requirements of the FORTRAN language. The restrictions are:

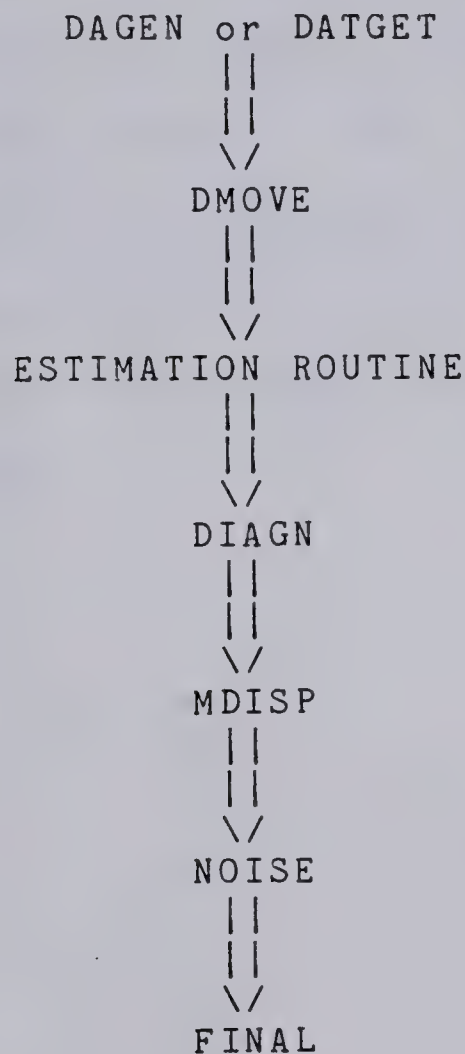
1) the amount of data is restricted to 550 input/output pairs

2) the model order of the process must be between one and five for STIME {3.9}, IVOFF {3.11} and UBIAS {3.12} and must be between one and three for ASTRM {3.10}

3) the time delay must be greater than or equal to zero

All programs are written in FORTRAN IV. There are seventeen major programs all interlinked by a steering

program, STEER {3.3}, which provides the user with the ability to control program flow. Each time program control is transferred to a new major module, the program's name is written and underlined. Normally, program flow is automatic and the user does not need to take personal step by step control. A flow diagram of the "normal" mode of control is:



Two options of note are:

- 1) the MONTE CARLO facility which allows the user to run the same data with the same model order and time delay range through all the estimation routines without having to go through the entire cycle, that is the

subroutines MDISP {3.14}, NOISE {3.15}, FINAL {3.16} and FDISP {3.17} are skipped.

- 2) the subroutine CRCOR {3.8} computes the cross correlation function between the modified input/output sequences. If the input sequence is a pseudo random binary sequence then an unbiased estimate of the weighting sequence is produced. If the input sequence is not a pseudo random binary sequence the cross correlation function will have the correct shape but the amplitudes will be incorrect. Thus correlation analysis is provided by UNALIP but this facility is not examined in this thesis.

PROGRAM MODULES

3.2 OXIP

PURPOSE: the mainline program. OXIP initializes all major system variables and transfers control of program flow to the subroutine STEER, {3.3}. The flow diagram for the OXIP module is figure 3-1.

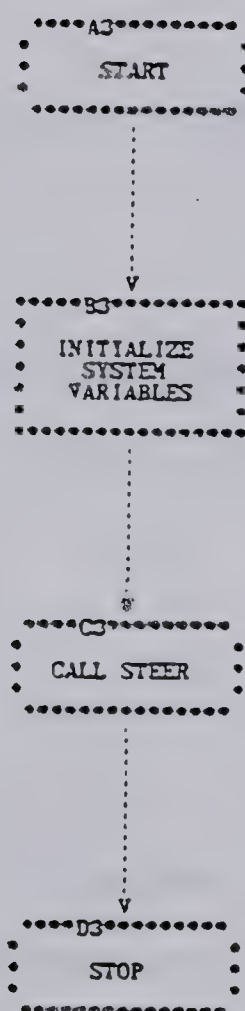


Figure 3-1 Flow Diagram for OXIP

VARIABLES:

A, B, C, D real vectors of length five used to store the coefficients of the A, B, C, and D polynomials.

COVM the covariance matrix used to store the matrix product $(\underline{X}^T \underline{X})^{-1}$

COVV the covariance vector used to store the vector product $\underline{X}^T \underline{y}$

E a real vector used to store the noise added to the unmodified output data in the subroutine DAGEN {3.6}

E1 a real vector used to store the residuals calculated in the estimation routine ie.) $E1 = Y1 - Y2$ where Y2 is an intermediate calculation of the system's output

E2 a real vector used to store the prediction errors (final residuals) calculated in the estimation routines ie.) $E2 = Y1 - Y2$

IFILE an integer variable that contains the logical unit number of an output file used to store calculated results from many subroutines

IFILE1 an integer variable that contains the logical unit number of an output file used to store the real vectors E, E1, E2, U, U1, U2, Y, Y1 and Y2

IN an integer variable that contains the logical unit number of the input device

IOUT an integer variable that contains the logical unit number of the output device

ISWCH an integer variable used as the major steering parameter that dictates program flow in STEER {3.3} and other subroutines

MONTE an integer variable used as a switch to indicate if the Monte Carlo facility is needed

RO a real vector that stores the parameters of an autoregressive model of the residuals

U a real vector used to store unmodified input data

U1 a real vector used to store input data modified by the subroutine DMOVE {3.7}

U2 a real vector used to store modified input data that is manipulated by the estimation routines

- Y a real vector used to store unmodified output data
- Y1 a real vector used to store output data modified by the subroutine DMOVE {3.7}
- Y2 a real vector used to store the prediction of the system output by the estimation routines

SUBROUTINES CALLED:

STEER {3.3} allows linkage to the rest of the
subroutines

3.3 SUBROUTINE: STEER

PURPOSE: a general steering program that regulates the flow of execution of UNALIP. The flow diagrams for the STEER module are figures 3-2, 3-3 and 3-4.

METHOD: the flow of execution is determined by the system variables ISWCH, JDAT, JSWCH and MONTE. These variables are set by different subroutines or by the user. This subroutine also sets up Monte Carlo runs and allows the user to exit from the package.

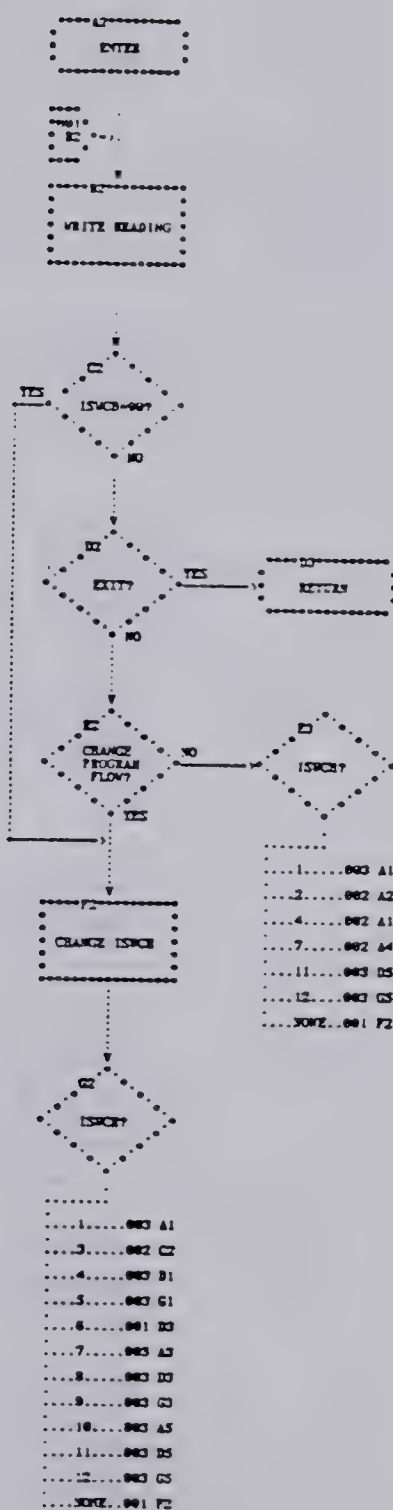


Figure 3-2 Flow Diagram for STEER

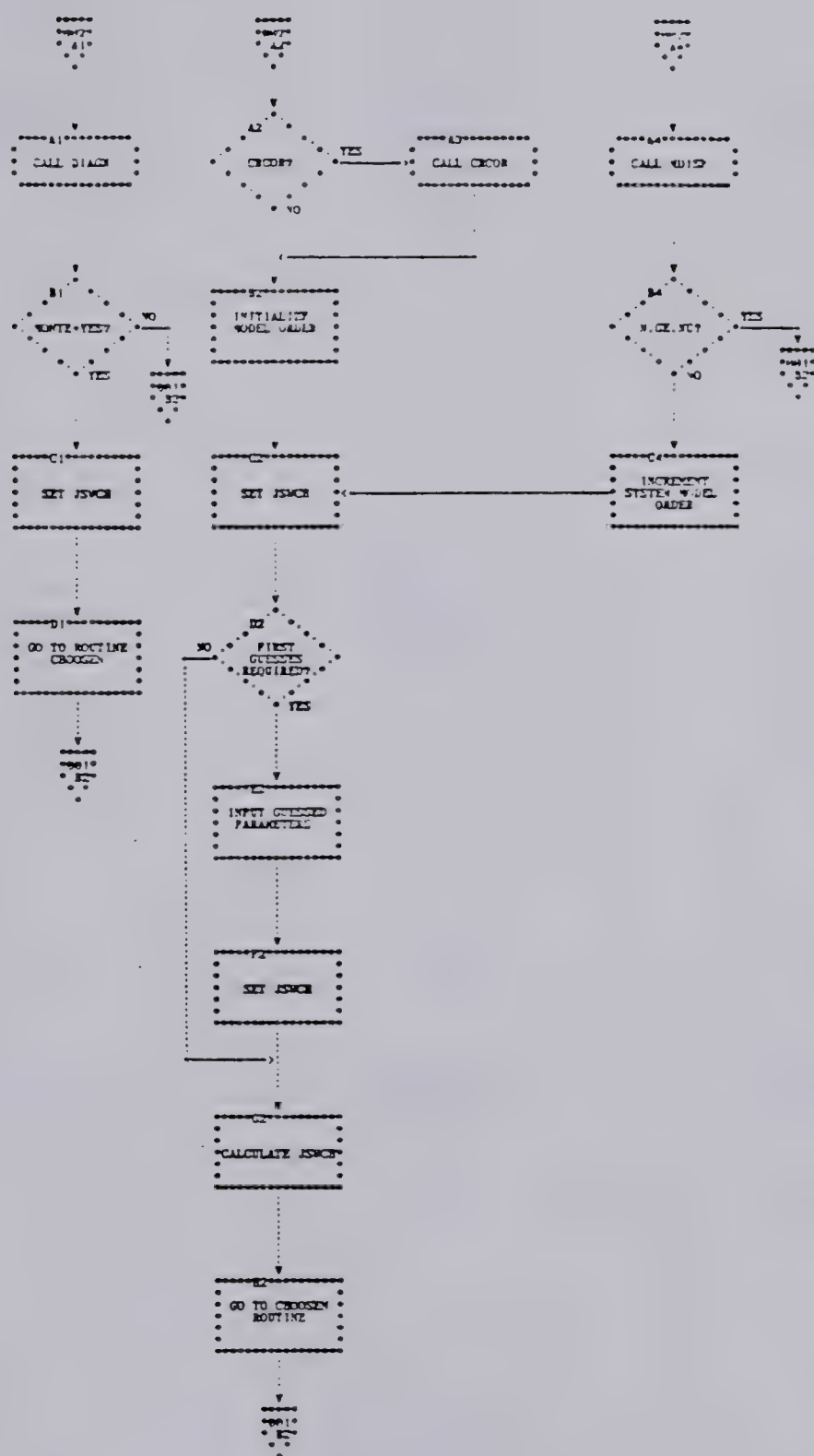


Figure 3-3 Flow Diagram for STEER

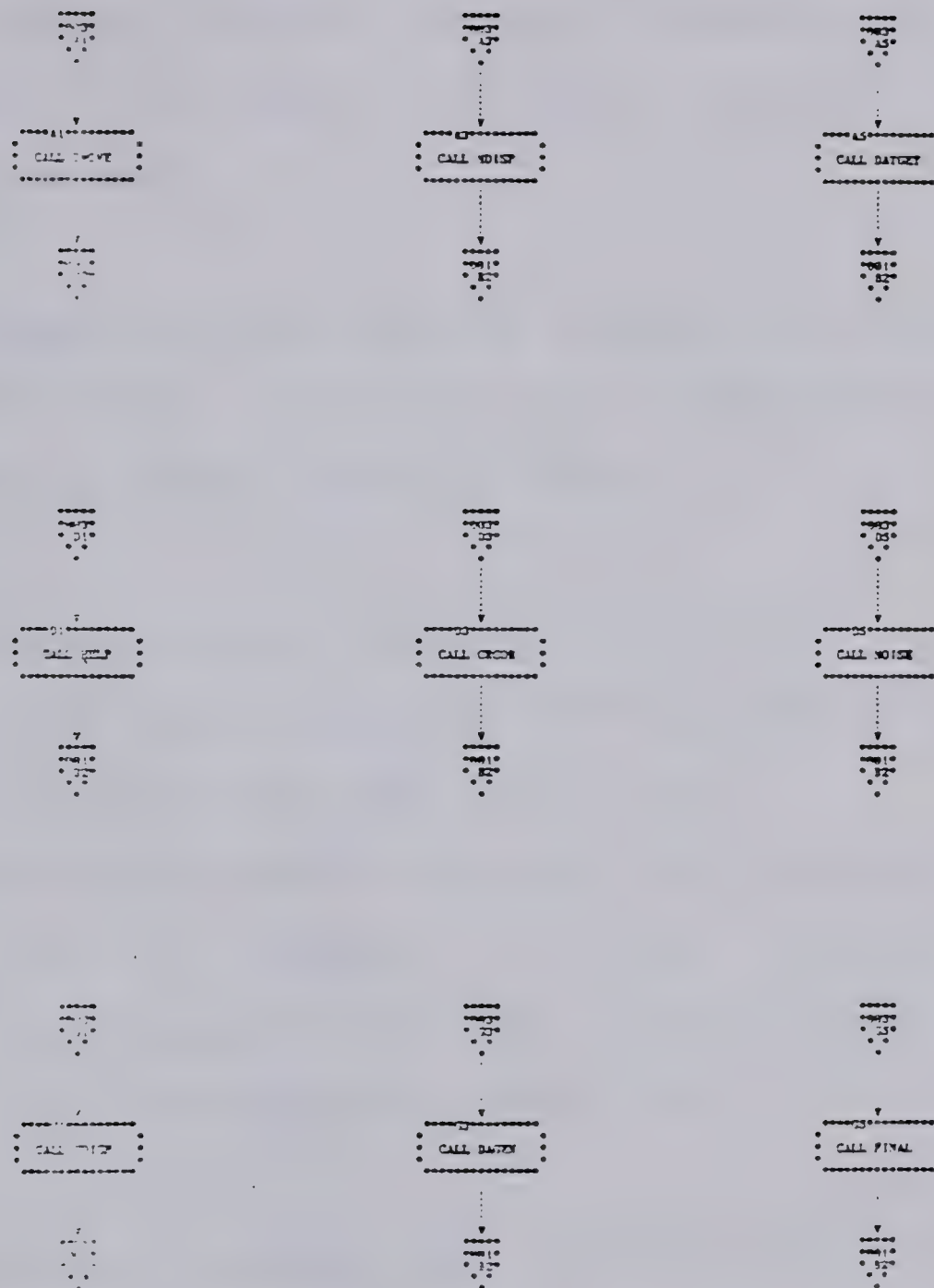


Figure 3-4 Flow Diagram for STEER

VARIABLES:

IRO an integer variable that contains the current autoregressive model order of the residuals

J an integer variable used in computed GO TO statements to determine which estimation routine is to be used

JDAT an integer variable used as a switch to indicate the direction of program flow determined in the minor subroutines DGSW and DGSW1

JSWCH an integer variable used as a switch in:

- 1) the minor subroutine PARSP to determine the type of calculation to be done
- 2) the estimation routines to determine if first guesses of the polynomial coefficients and other variables were used in the subroutine STEER {3.3}

N an integer variable used to hold the current system model order

NL integer variable used to hold the lower bound of the system model orders to be investigated

NU integer variable used to hold the upper bound of
the system model orders to be investigated

YES an integer variable used to indicate program flow
determined by the minor subroutines DGSW and
DGSW1

SUBROUTINES CALLED:

ASTRM {3.10} an estimation routine using the maximum likelihood algorithm of Astrom et al [3]

CRCOR {3.8} computes the cross correlation function between the modified input and the modified output data

DAGEN {3.6} generates output data for a noisy, discrete time system excited by a predetermined input sequence

DATGET {3.5} transfers input/output data from user files to system vectors

DGSW supplies messages and program flow control at decision points in the program

DIAGN {3.13} takes the current calculated system and performs statistical tests to determine the 'goodness of fit' with the real system

DMOVE {3.7} modifies raw input/output data and certain system variables are also obtained here

FDISP {3.17} supplies graphical displays of data files

FINAL {3.16} gives various tabulated results for the system considered

FREAD allows the user to use free format input and provides certain error checks for the input [21]

HELP {3.4} describes various input variables required by the major program modules

IVOFF {3.11} an estimation routine using an off-line (non-recursive) instrumental variable method

MDISP {3.14} allows the user to determine the impulse response of the optimum system model

NOISE {3.15} estimates noise models from uncorrelated measurement error data from the subroutine MDISP {3.14}

PRINT writes out repetitive headings

STIME {3.9} an estimation routine using the generalized least squares algorithm of Clarke [7]

UBIAS {3.12} an estimation routine using an unbiased algorithm [16] based on Slutsky's Theorem

3.4 SUBROUTINE: HELP

PURPOSE: to describe the various input variables required by the major program modules. The flow diagram for the HELP module is figure 3-5.

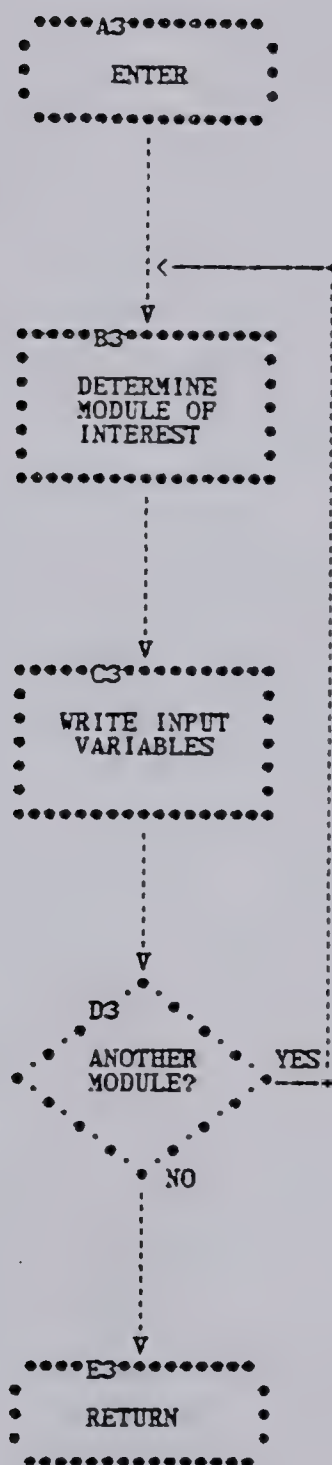


Figure 3-5 Flow Diagram for HELP

VARIABLES:

KSWCH an integer variable used to determine program
flow

SUBROUTINES CALLED:

FREAD allows the user to use free format input and provides certain error checks for the input [21]

3.5 SUBROUTINE: DATGET

PURPOSE: to transfer input/output data from user files to system vectors. The flow diagram for the DATGET module is figure 3-6.

METHOD: the input/output data is stored in two files DATIN and DATOUT that are assigned the FORTRAN logical unit numbers 18 and 19. The minor subroutine FREAD [21] is used to transfer the data from the files to the system vectors U and Y.

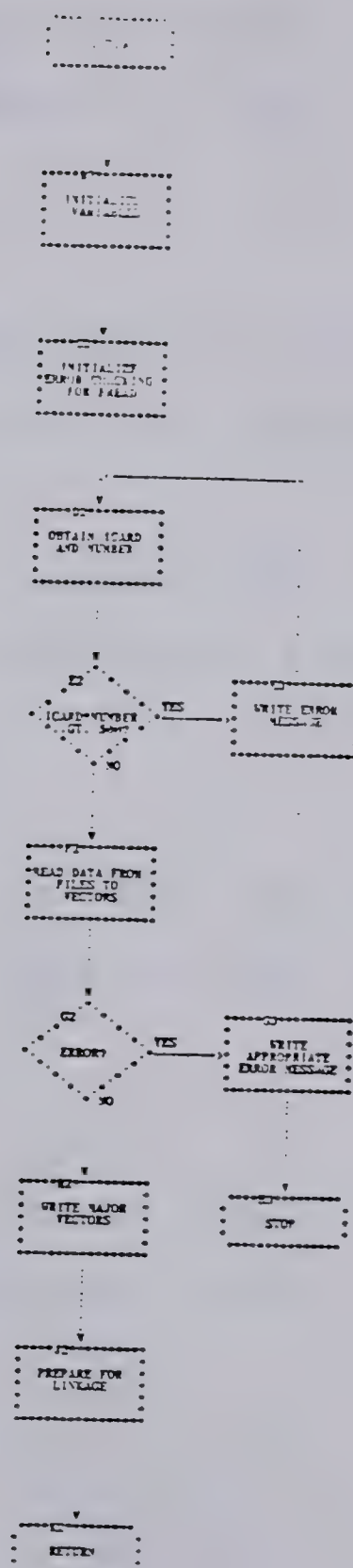


Figure 3-6 Flow Diagram for DATGET

VARIABLES:

ICARD an integer variable that holds the number of
 lines in the input/output data files

INDAT, IOUTDA integer variables that contain the
 logical unit numbers of the input/output data
 sources

JJ an integer variable that is used to transfer the
 data from the files to the vectors

LENGTH an integer variable that contains the total
 number of input/output pairs transferred from the
 files

NUMBER an integer variable that holds the number
 of values on a line of the input/output data
 files

USAVE, YSAVE real vectors used to store one line
 of the input/output data files

SUBROUTINES CALLED:

FREAD allows the user to use free format input and provides certain error checks for the input [21]

3.6 SUBROUTINE: DAGEN

PURPOSE: to generate output data for a noisy, discrete time system excited by a predetermined input sequence. The flow diagrams for the DAGEN module are figures 3-8 and 3-9 and the box diagram for the process is figure 3-7.

METHOD: the output data is generated by the model

$$\underline{y} = \frac{z^{-k} BD(z^{-1}) \underline{u}}{[1 + AD(z^{-1})](1 - z^{-1})^{IDRS}} + \frac{[1 + DD(z^{-1})] \underline{w}}{[1 + CD(z^{-1})](1 - z^{-1})^{IDRN}} \quad (3.2)$$

where: 1) \underline{u} is a predetermined input sequence

2) z is the z -transform variable

$$3) \quad BD(z^{-1}) = bd_1 z^{-1} + \dots + bd_{ND} z^{-ND}$$

$$4) \quad AD(z^{-1}) = ad_1 z^{-1} + \dots + ad_{ND} z^{-ND}$$

$$5) \quad DD(z^{-1}) = dd_1 z^{-1} + \dots + dd_{NOD} z^{-NOD}$$

$$6) \quad CD(z^{-1}) = cd_1 z^{-1} + \dots + cd_{NOD} z^{-NOD}$$

7) \underline{w} is a normally distributed uncorrelated random sequence (white noise)

8) \underline{y} is the resultant output data sequence

- 9) bd_i , ad_i , dd_i and cd_i are the system parameters supplied by the user
- 10) IDRS is the number of integrations to be performed on the output sequence
- 11) IDRN is the number of integrations to be performed on the noise sequence

BOX DIAGRAM FOR DAGEN

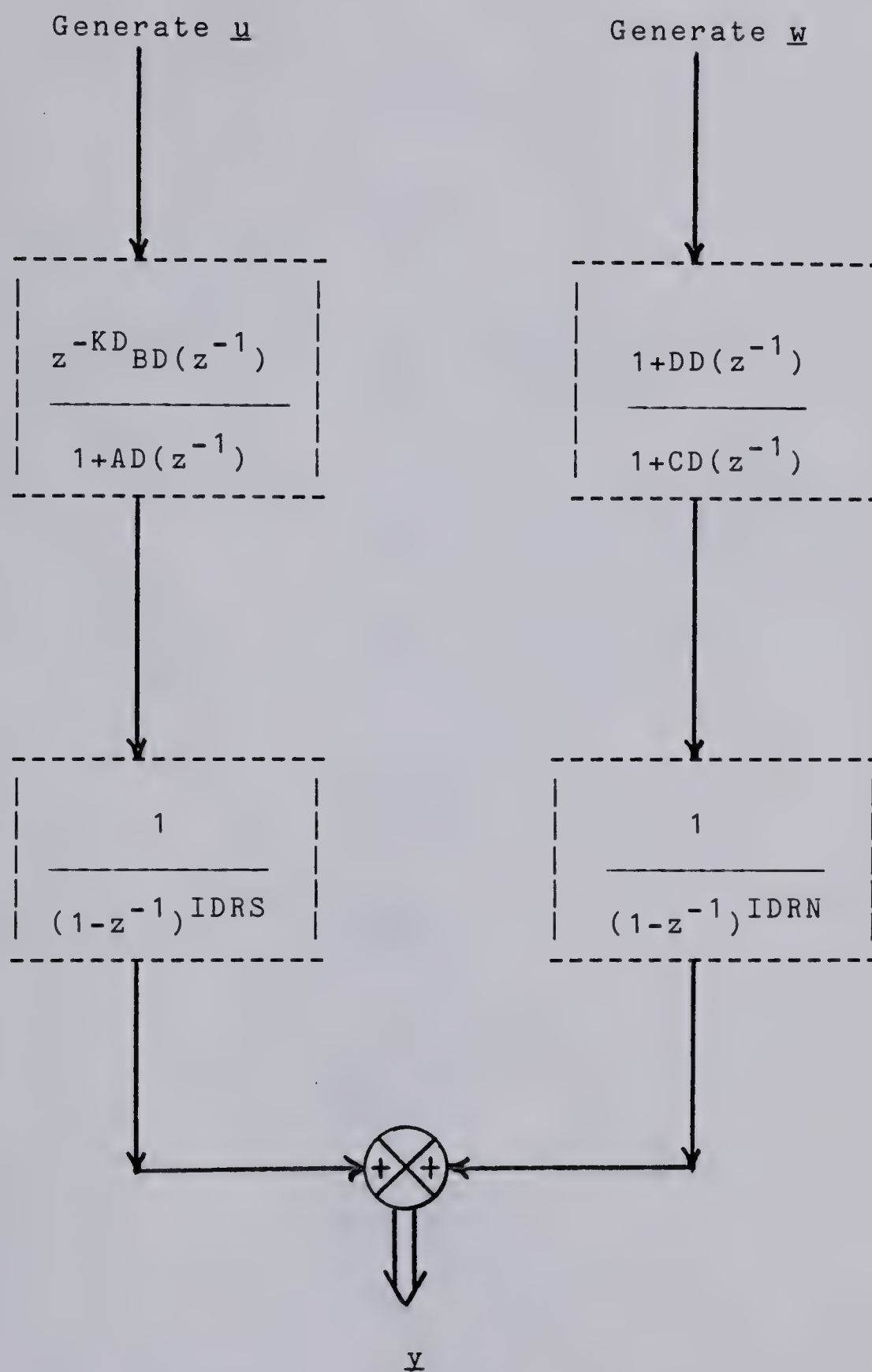


Figure 3-7

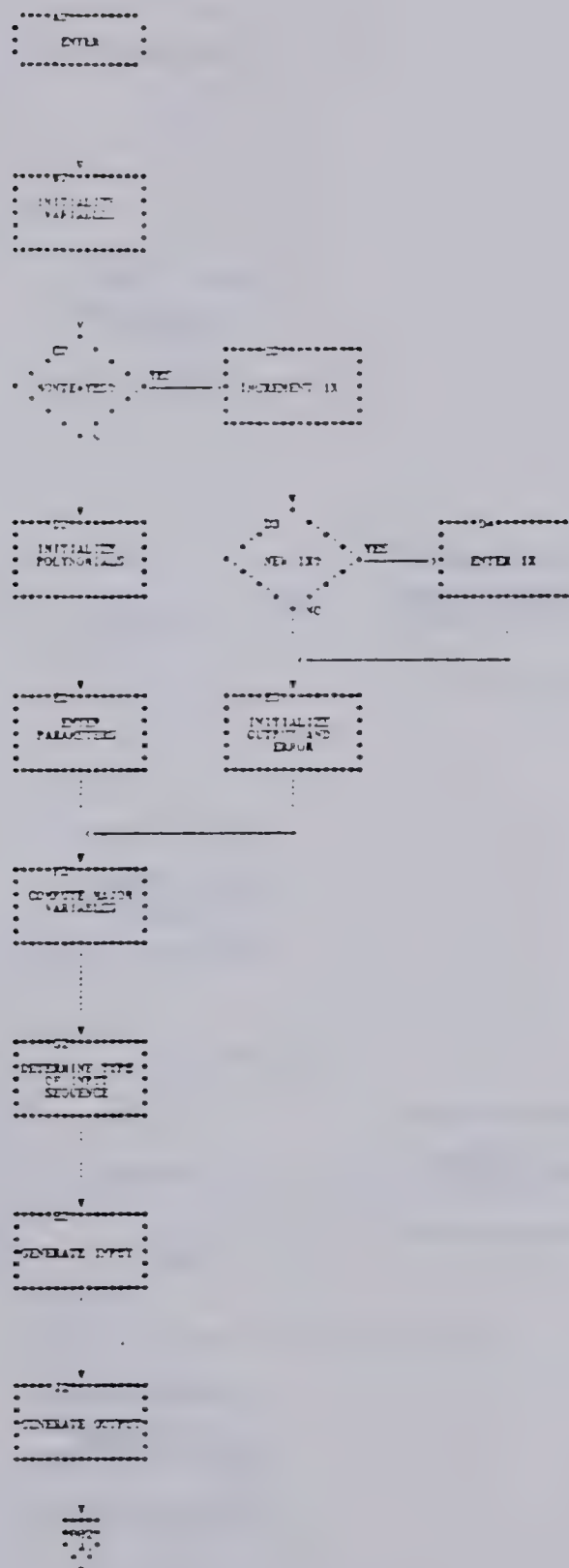


Figure 3-8 Flow Diagram for DAGEN

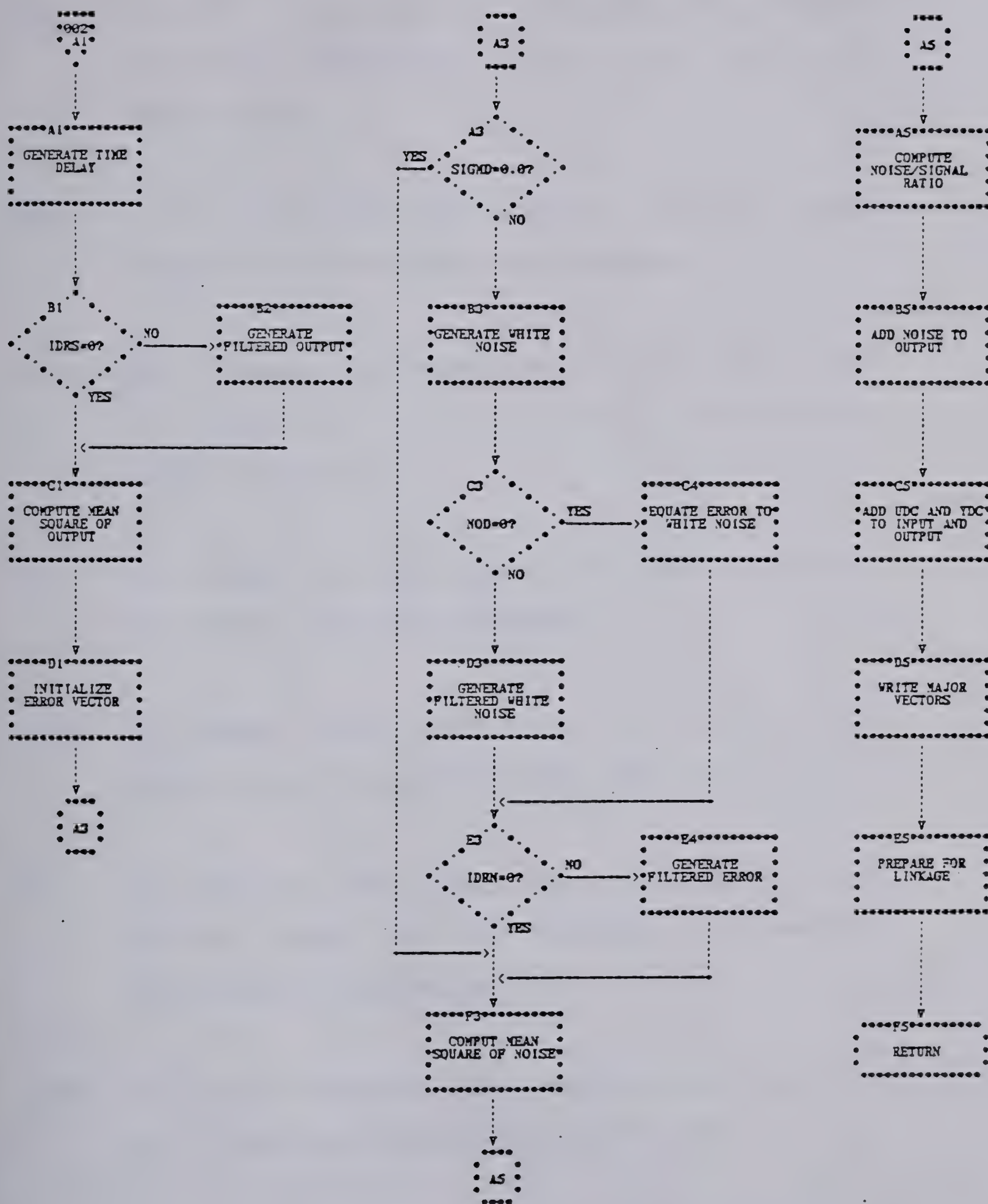


Figure 3-9 Flow Diagram for DAGEN

VARIABLES:

AD, BD, CD, DD real vectors of length five used to store the coefficients of the AD, BD, CD, and DD polynomials

EMS a real variable that contains the mean square value of the generated noise sequence

IDRN, IDRS integer variables that contain the number of integrations to be performed on the noise and output sequences

IJ an integer variable used in DO loops to compute the output and noise sequences

IOUTP an integer vector that holds the output of the pseudo random binary sequence generator

IX an non-zero odd integer variable that is the initial seed of the Gaussian and uniform distribution generators [19]

JFRST an integer variable that contains the number of the first input/output pair considered

JLAST an integer variable that contains the number of the last input/output pair considered

- JN 1) an integer variable used as an initial value in a DO loop to calculate $\underline{y} = BD(z^{-1})\underline{u} / [1 + AD(z^{-1})]$
- 2) an integer variable used as an initial value in a DO loop to calculate $\underline{\varepsilon} = [1 + DD(z^{-1})\underline{w}] / [1 + CD(z^{-1})]$
- JTOT an integer variable that contains the number of input/output pairs generated plus the number of time delays plus one
- JX an integer variable that is initially equated to IX so that IX is not destroyed by the minor subroutines GAUSS and RANDU
- KD an integer variable that contains the number of time delays in the generated system
- KX an integer variable used in the minor subroutine RANDU to return the next value of the seed for the generator
- NCODE an integer variable that determines the length of the psuedo random binary sequence
ie.) $LENGTH = 2 * NCODE - 1$

ND an integer variable that contains the generated system model order

NOD an integer variable that contains the generated noise model order

NTOTD an integer variable that contains the number of input/output pairs generated

PERID a real variable that contains the period (usually in multiples of the sample time) of the sinusoidal disturbance that generates U

RAND a real variable used to store the white noise sequence

S a real variable used to store intermediate calculations of Y and E

SIGND a real variable that contains the standard deviation of the Gaussian distribution generated to give w

UDC, YDC real variables used to add constant values to all elements of the input and output sequences generated

YMS a real variable that contains the mean square
value of the generated output sequence

ZNSR a real variable that contains the noise/signal
root mean square ratio

SUBROUTINES CALLED:

DGSW supplies messages and program flow control at
 decision points in the program

FREAD allows the user to use free format input and
 provides certain error checks for the input [21]

GAUSS generates a normally distributed uncorrelated
 random sequence (white noise) [19]

PRBS generates a pseudo random binary sequence of
 variable length

RANDU generates a uniformly distributed random sequence
 [19]

3.7 SUBROUTINE: DMOVE

PURPOSE: to enter parameters required by estimation routines and to perform transformation of the raw data to mean free modified data. The flow diagram for the DMOVE module is figure 3-11 and the box diagram for the process is figure 3-10.

METHOD: the data is transformed by differencing and high pass filters

$$\underline{y}' = \underline{y}(1-z^{-1})^{IYDIF} \frac{(1-z^{-1})^{IDRY}}{(1-TCU*z^{-1})}$$

$$\underline{u}' = \underline{u}(1-z^{-1})^{IUDIF} \frac{(1-z^{-1})^{IDRU}}{(1-TCU*z^{-1})}$$

where: 1) \underline{y} , \underline{u} are the raw input/output data

2) \underline{y}' , \underline{u}' are the modified input/output data

3) z is the z -transform variable

BOX DIAGRAM FOR DMOVE

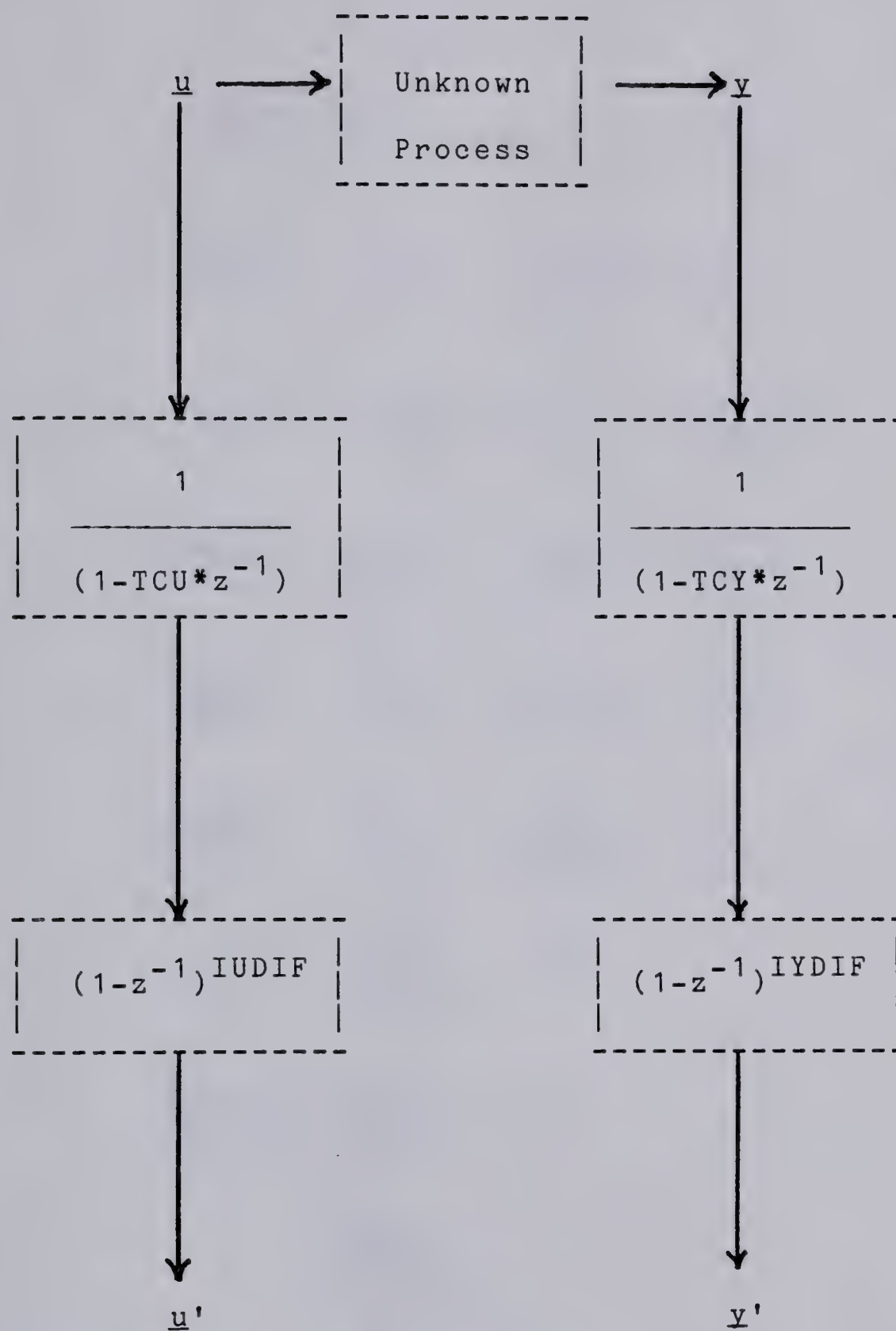


Figure 3-10

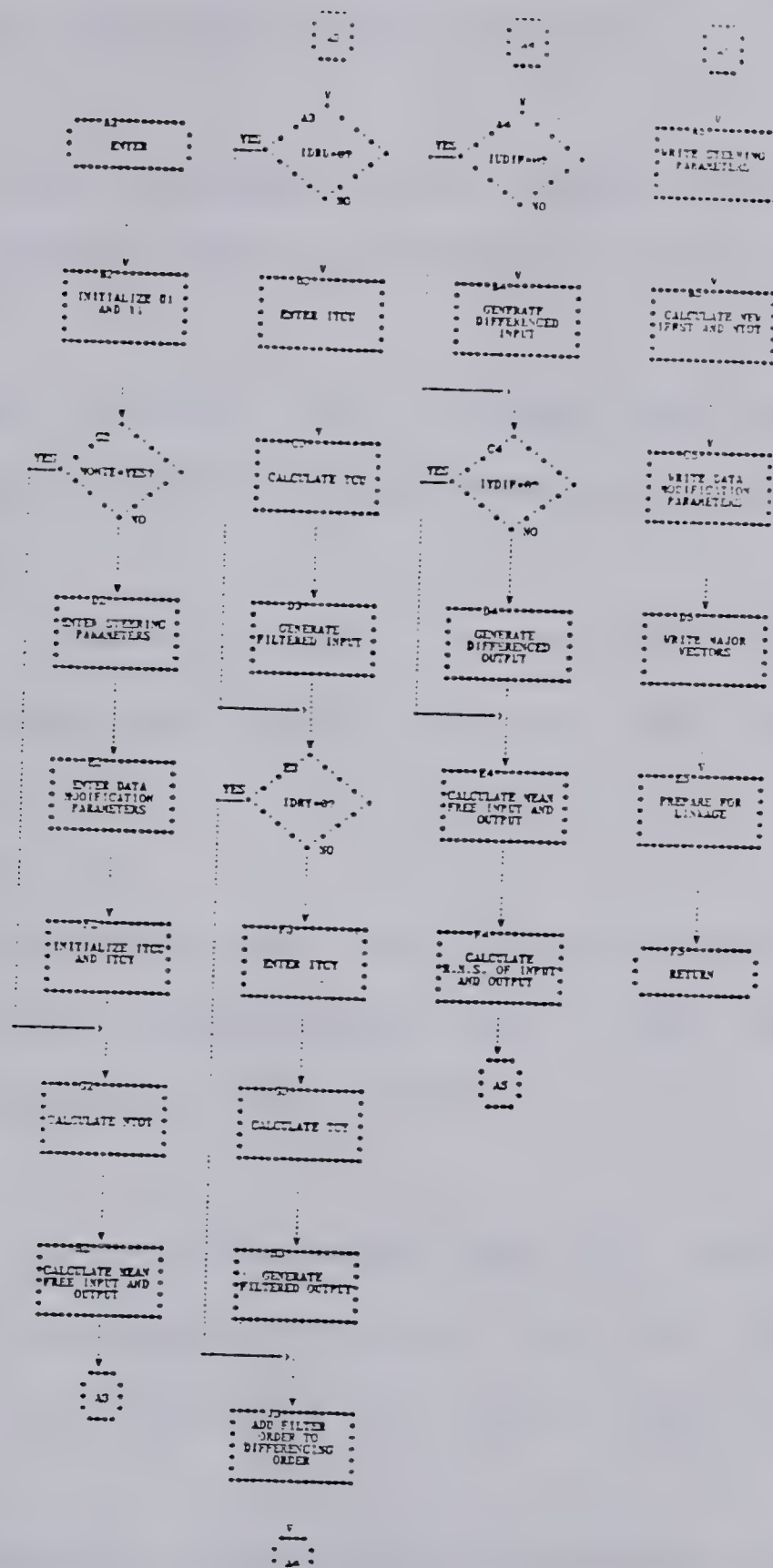


Figure 3-11 Flow Diagram for DMOVE

VARIABLES:

AU, AY real variables that contain the means of
the raw input and output sequences

AUU, AYY real variables that contain the means of
the modified input and output sequences

CRITE a real variable that contains the convergence
criterion used in the estimation routines

IDRU, IDRY integer variables that contain the order of
the high pass filter for the input and output
sequences

IFRST an integer variable that contains the number of
the first input/output pair to be used by the
estimation and other routines

ITCU, ITCY integer variables used to calculate the
time constants (s-space) for the high pass
filters of the input and output sequences

ITMAX an integer variable that contains the maximum
number of iterations used in the estimation
routines

IUDIF, IYDIF integer variables that contain the number

of differencings of the input and output sequences

JFRST1 an integer variable that contains the number of the first input/output pair to be modified by the subroutine DMOVE {3.7}

KFRST an integer variable that is used to initialize the first values of a data sequence that is to be operated on by a low pass filter

KL an integer variable used to hold the lower bound of the time delays to be considered

KU an integer variable used to hold the upper bound of the time delays to be considered

LAST an integer variable that is used to hold the number of the last data pair to be used by the estimation and other routines

RMSU, RMSY real variables that contain the root mean squared values of the modified input and output sequences

TCU, TCY real variables that contain the time constants for the high pass filters of the input

and output sequences eg.) $TC = \exp(-T/RC)$

where: 1) T is the sampling interval (equated to
unity for this package)

2) RC is the time constant of the filter
(equated to ITC)

UMEAN, YMEAN real variables that contain the sum of the
means of the raw and modified input and output
sequences

SUBROUTINES CALLED:

FREAD allows the user to use free format input and provides certain error checks for the input [21]

3.8 SUBROUTINE: CRCOR

PURPOSE: to compute the cross correlation function between the modified input and modified output. The flow diagram for the CRCOR module is figure 3-12.

METHOD: if the modified input sequence is a pseudo random binary sequence, the scaling is chosen to result in an unbiased estimate of the system's weighting sequence by the cross correlation function. If the modified input sequence is not a psuedo random binary sequence, the cross correlation function will have the correct shape of the weighting sequence but the amplitudes will be wrong. The mean of the weighting sequence is removed from the sequence and a graph of the cross correlation function can be obtained.

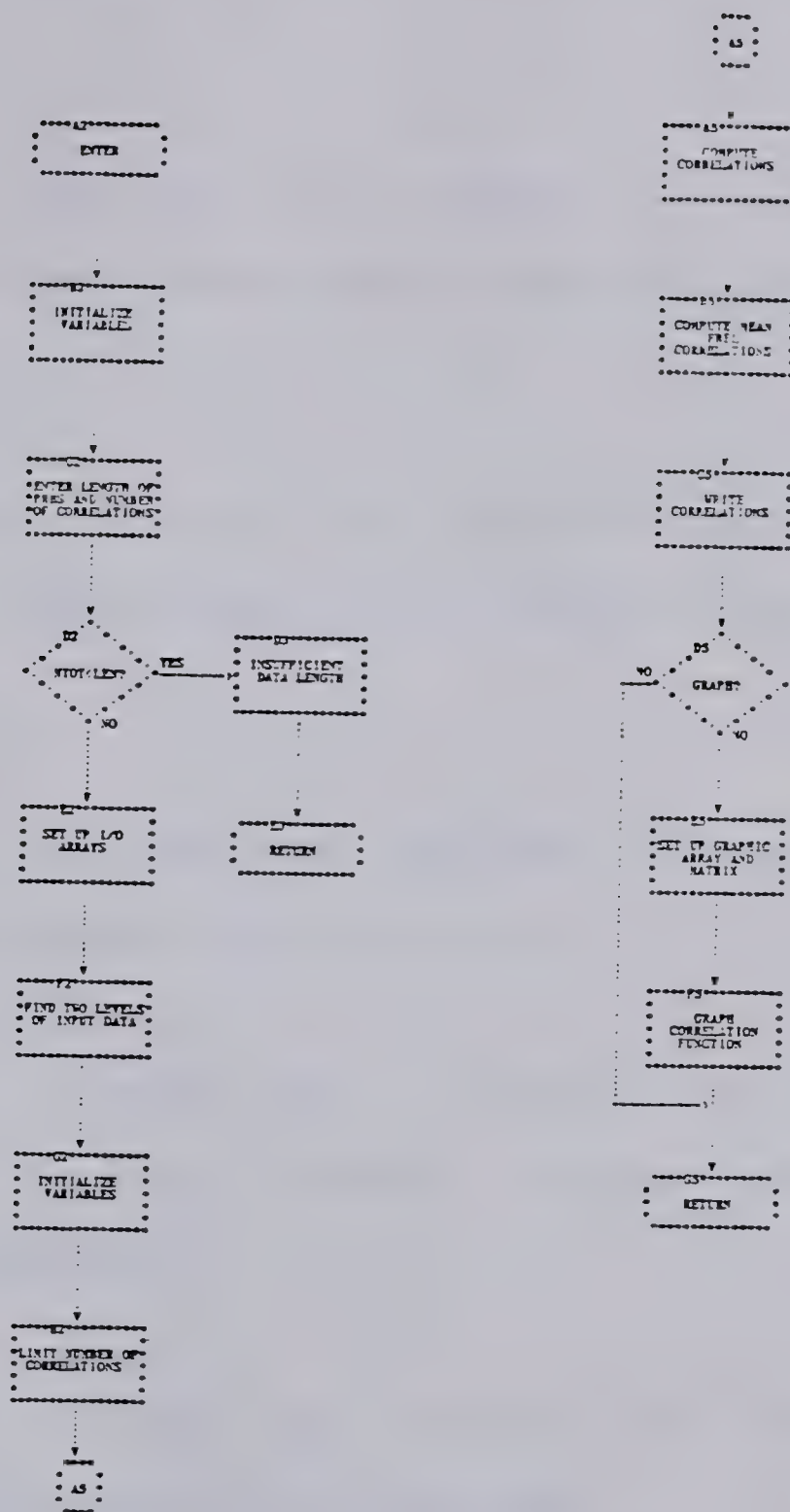


Figure 3-12 Flow Diagram for CRCOR

VARIABLES:

- INOC an integer variable that contains the number of times the pseudo random binary sequence is repeated
- JLAST an integer variable that contains the number of the last input/output pair considered by this program
- JTOTC an integer variable that contains the total number of input/output pairs considered by this program
- LEN an integer variable that contains the length of the pseudo random binary sequence
- NCODE an integer variable that determines the length of the pseudo random binary sequence
ie.) $LEN = 2^{NCODE} - 1$
- NCORR an integer variable that contains the number of cross correlations to be performed
- S 1) a real variable used to determine the two different levels in the modified pseudo random binary sequence

2) a real variable used to determine the mean of
the cross correlation function

SCAL a real variable that contains the scaling factor
for the cross correlation function

T a real variable that contains the difference in
the two levels of the modified pseudo random
binary sequence

UDISK a real vector that holds the cross correlation
function

UU, YY real vectors that contain the input and
output sequences used to generate the cross
correlation function

XX, ZZ a real vector and array that are used to
pass functions to the minor subroutine GRAPH

SUBROUTINES CALLED:

DGSW supplies messages and program flow control at decision points in the program

FREAD allows the user to use free format input and provides certain error checks for the input [21]

GRAPH allows the user to plot on a hard copy terminal, a graph of a function supplied by the user

3.9 SUBROUTINE: STIME

PURPOSE: to calculate the coefficients of a model of given order, choosing the best time delay over the allowed range, using the generalized least square method of Clarke [7]. The flow diagrams for the STIME module are figures 3-13, 3-14 and 3-15.

METHOD: unbiased parameter estimates of a noisy process using a least squares solution can be obtained by filtering the input/output sequences using a moving average filter. The coefficients of the moving average filter are the coefficients of the autoregressive model of the noise process. This leads to the estimate

$$\hat{\underline{\theta}}^* = [\underline{X}^{*T} \underline{X}^*]^{-1} \underline{X}^{*T} \underline{y}^* \quad (3.3)$$

where: 1) \underline{X}^* is the filtered result of the matrix \underline{X}

- 2) $\hat{\underline{\theta}}^*$ is the estimate of the parameter vector $\underline{\theta}$ for the filtered data
- 3) \underline{y}^* is the filtered corrupted output of the system

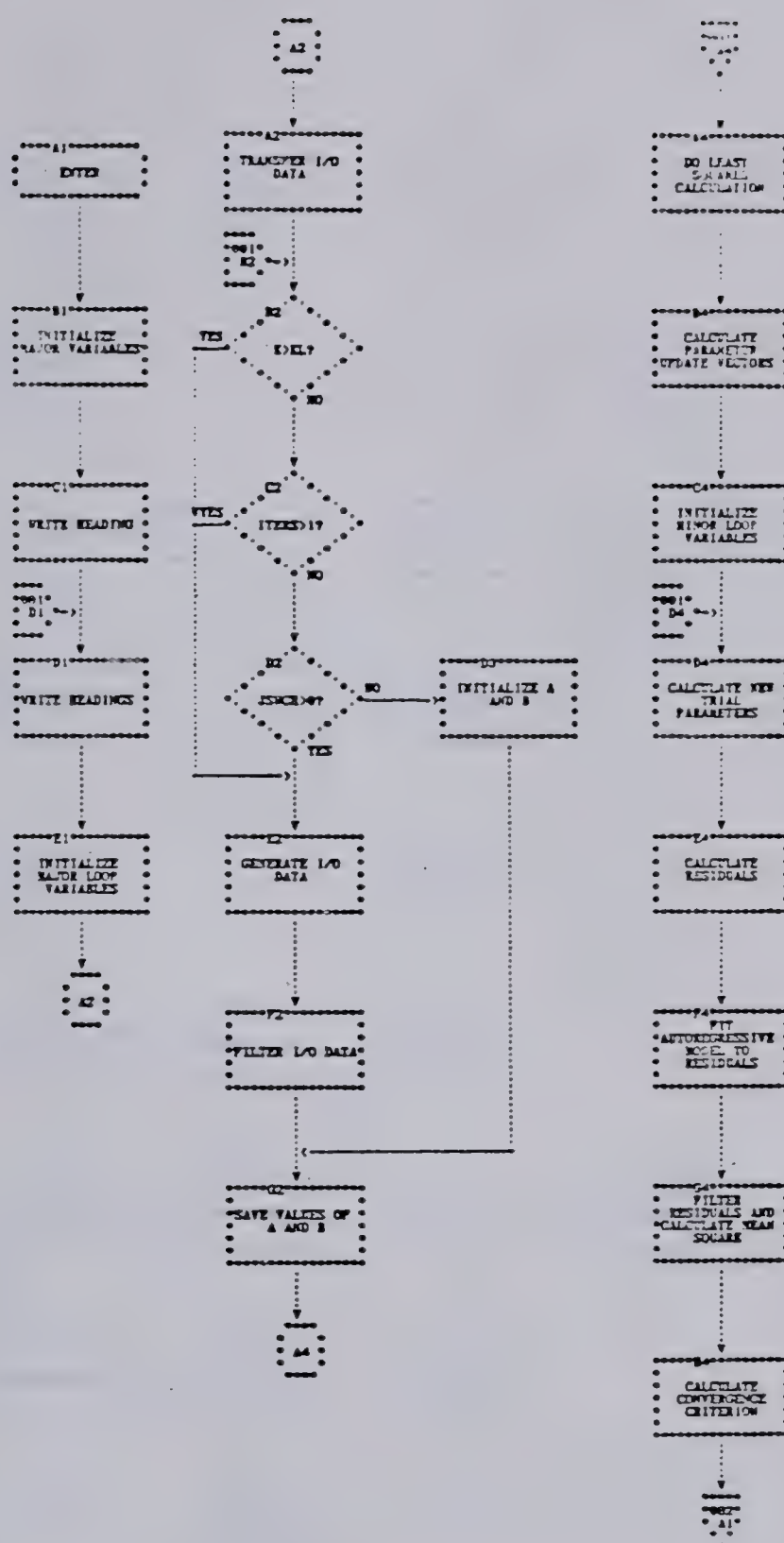


Figure 3-13 Flow Diagram for STIME

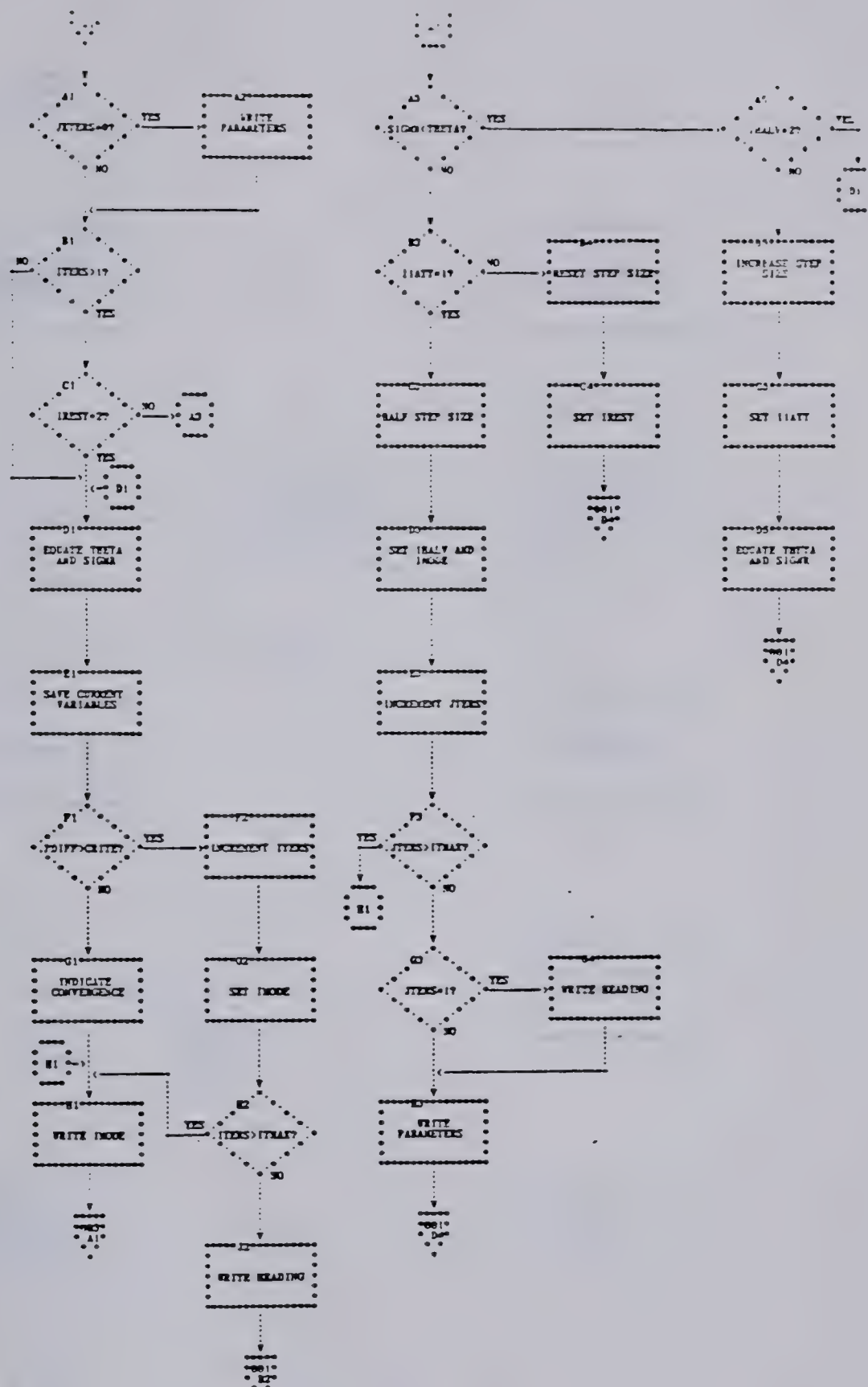


Figure 3-14 Flow Diagram for STIME

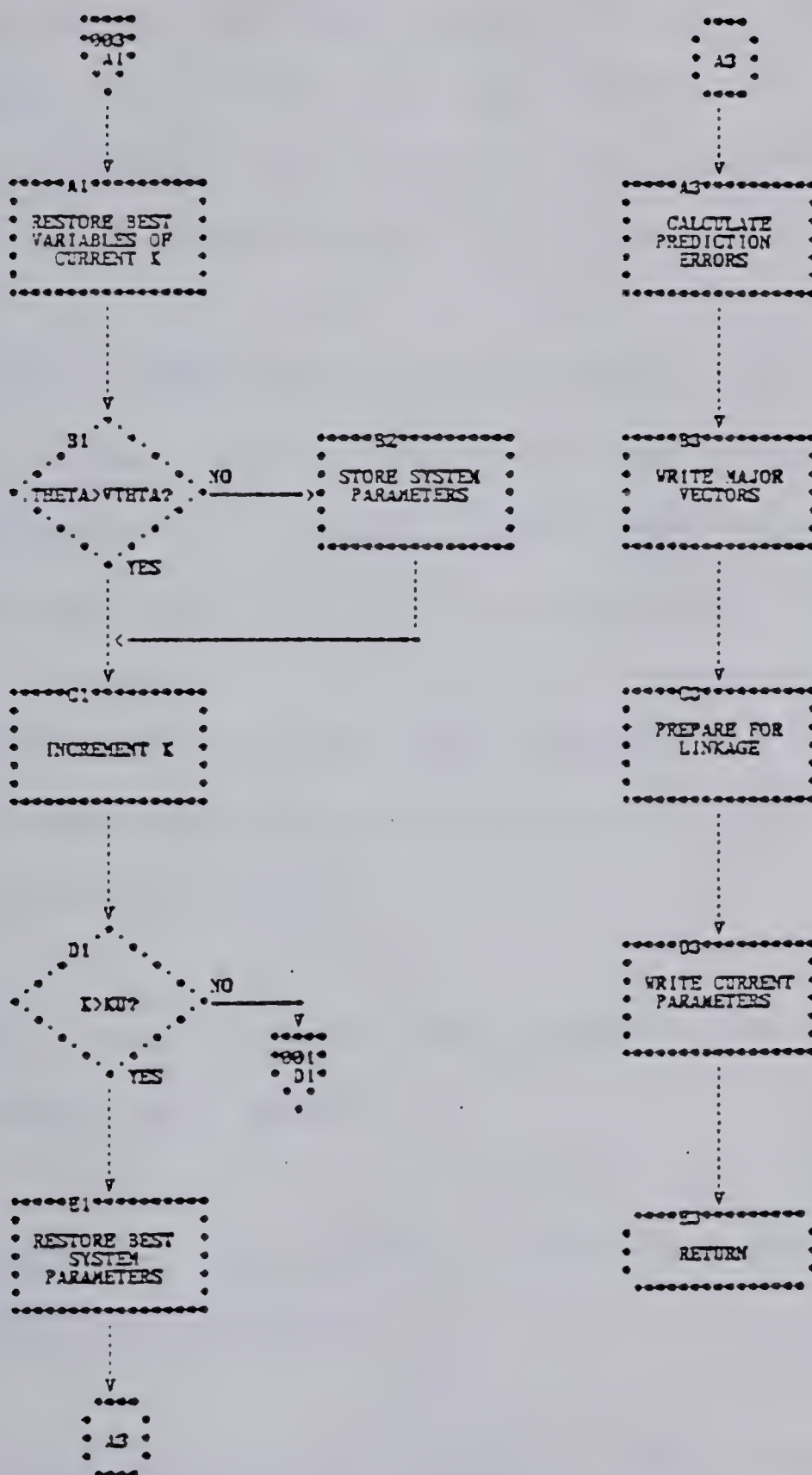


Figure 3-15 Flow Diagram for STIME

VARIABLES:

AA, BB, CCOVM, DET, EE, IIRO, KK, RRO, SIGM, UU, YY are real and integer variables, vectors and arrays that store the best values of A, B, COVM, COVV, DETE, E1, IRO, K, RO, SIGMA, U2 AND Y2 respectively for the model under consideration by the estimation routine

A1, B1, CKOVM, CKOVV, DDETE, ROR, SSIGMR are real variables, vectors and arrays that store the best values of A, B, COVM, COVV, DETE, RO and SIGMA for each iteration of the estimation routine

ADIFF, BDIFF real vectors that contain the difference between the previous iteration and least squares estimate of A and B

ANEW, BNEW real vectors that contain the parameter updates of A and B

ASAVE, BSAVE real vectors that contain A and B from the previous iteration

DETE a real variable that contains the determinant of the covariance matrix

DELUP a real variable that contains the step size for the approximation of A and B

EAVE a real variable that contains the average of the residuals

IHALV an integer variable that indicates if the step size has been halved

IMODE an integer variable that indicates the type of exit from an estimation routine for a particular time delay

- 1 = convergence
- 2 = divergence
- 3 = ITMAX has been exceeded

IREST an integer variable that indicates if the step size has been reset to an earlier better value

ITERS an integer variable that contains the current number of iterations in the major loop of the estimation routine

IIATT an integer variable that indicates if the step size has been increased

JN an integer variable that contains the dimension
 of the square matrix COVM

JTERS an integer variable that limits the number of
 times the step size can be halved

K an integer variable that is used to hold the
 value of the current time delay being considered
 by the estimation routine

MODE an integer variable that indicates the type of
 exit from an estimation routine for the optimal
 time delay

 1 = convergence

 2 = divergence

 3 = ITMAX has been exceeded

PDIFF a real variable that contains the convergence
 criterion calculated by the estimation routine

R, S real variables used for intermediate calculations
 of ANEW and BNEW

SIGMR a real variable used to contain the mean square
 value of the residual

THETA a real variable used to determine if the current iteration is better than the last iteration

VTHTA a real variable used to determine which time delay gave the best results

SWITCH POSITIONS:

IREST = 1 initial position of the switch
= 2 indicates that the step size has been
returned to a value that gave better results

IIATT = 1 initial position of the switch
= 2 indicates that the step size has been
increased

IHALV = 1 initial position of the switch
= 2 indicates that the step size has been halved

SUBROUTINES CALLED:

AFILT filters an array using an autoregressive filter

FCURG computes the parameters of an autoregressive fixed order model

MCOMP computes the covariance matrix, the covariance vector and the determinant of the covariance matrix

PARSP prints out the current model order results

PREDS computes the deterministic predictions of the discrete model

PRINT prints out repetitive headings

RUNAV filters an array using a moving average filter

3.10 SUBROUTINE: ASTRM

PURPOSE: to calculate the coefficients of a model of given order, choosing the best time delay over the allowed range, using Astrom's et al maximum likelihood algorithm [3]. The flow diagrams for the ASTRM module are figures 3-16, 3-17 and 3-18.

METHOD: to maximize the likelihood function, the loss function

$$V(\underline{\theta}) = \frac{1}{2} \sum_{t=1}^N w_t^2 \quad (3.4)$$

is minimized by use of the modified Gauss-Newton-Marquardt algorithm, using a modified Levenburg parameter. Thus the parameter update equation is

$$\underline{\hat{\theta}}^{t+1} = \underline{\hat{\theta}}^t - [\Psi \underline{S}_{\underline{\hat{\theta}}}(\underline{\hat{\theta}}^t)]^{-1} \underline{S}_{\underline{\hat{\theta}}}(\underline{\hat{\theta}}^t) \quad (3.5)$$

where: 1) $\underline{\hat{\theta}}$ is the vector of estimated parameters $\underline{\hat{a}}$, $\underline{\hat{b}}$ and $\underline{\hat{f}}$
 2) $\underline{S}_{\underline{\hat{\theta}}}$ is the gradient of the loss function

- 3) $\hat{\underline{S}}_{\underline{\theta}\underline{\theta}}$ is the approximated matrix of second order partial derivatives of the loss function
- 4) ψ is the modified Levenberg parameter which multiplies the diagonal elements of $\hat{\underline{S}}_{\underline{\theta}\underline{\theta}}$ by a constant
- 5) \hat{w}_t is the approximated noise source of the model

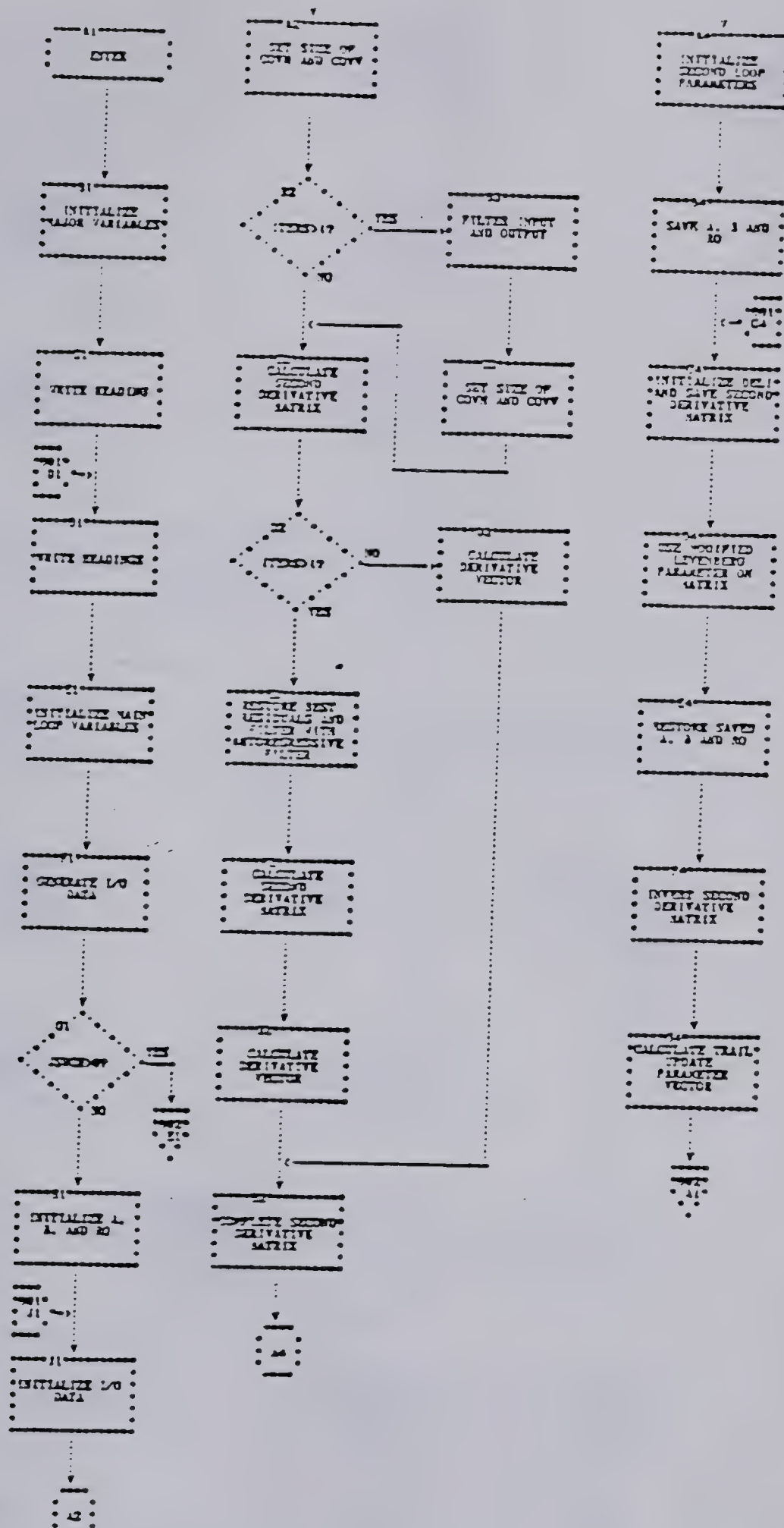


Figure 3-16 Flow Diagram for ASTRM

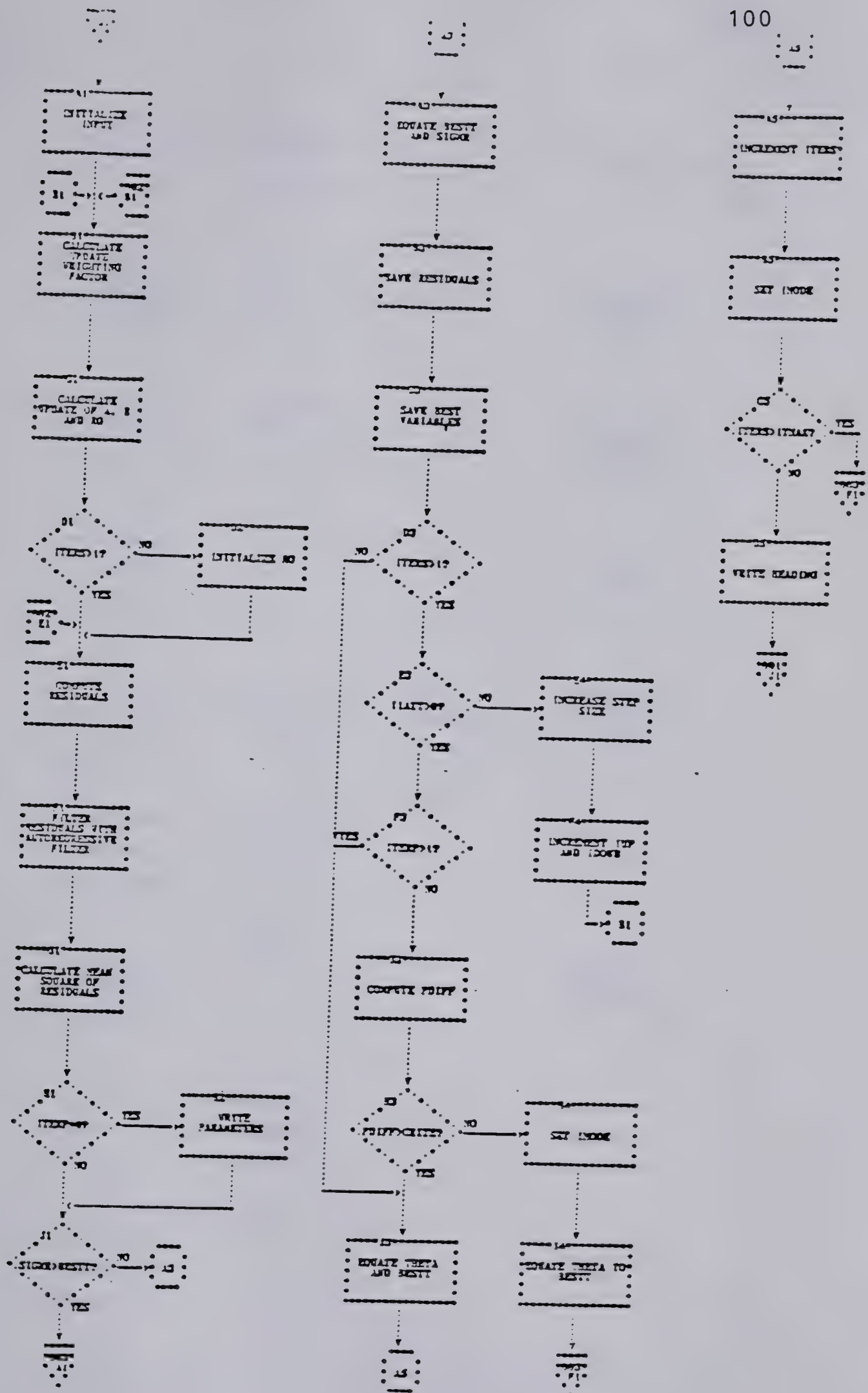


Figure 3-17 Flow Diagram for ASTRM

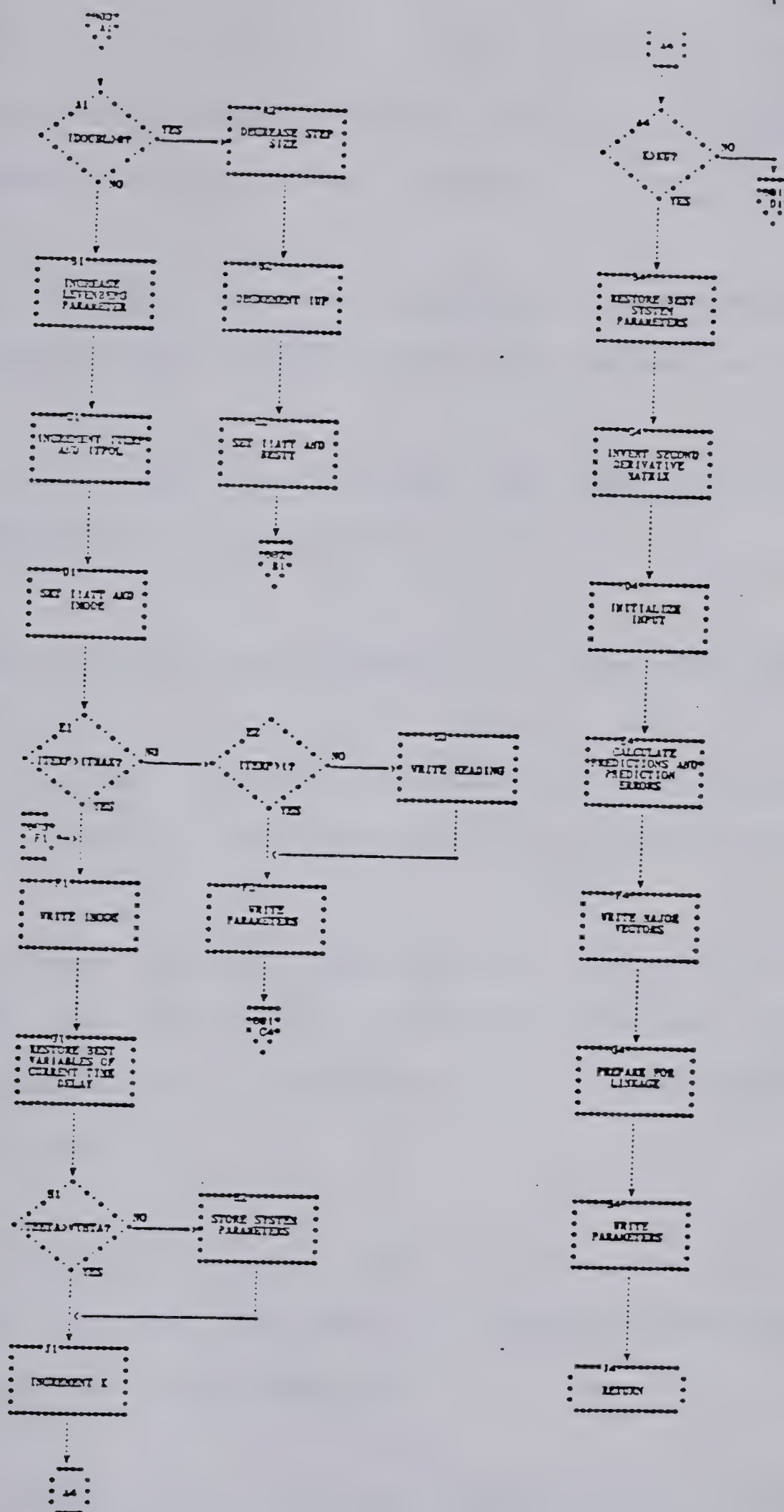


Figure 3-18 Flow Diagram for ASTRM

VARIABLES:

A2 a real vector used to contain the best residual sequence of a particular iteration

BESTT a real variable used to determine if the current value of SIGMR is better than the previous value

DEL a real variable that contains the step size for a change in the parameters

DELI a real variable used to modify the step size DEL

DELUPA a real variable that contains the first trial point for the step size of a new iteration

ITERP an integer variable that contains the number of times the modified Levenburg parameter is multiplied by a constant for a particular iteration

ITPOL an integer variable that contains the total number of times the modified Levenburg parameter is multiplied by a constant

IUP an integer variable that contains the total number of times the step size is successfully doubled

- JJN an integer variable that contains the dimension of the approximated second order partial derivative matrix
- MSIZE an integer variable that contains the current dimension of the square matrix COVM
- RLEVN a real variable containing the modified Levenburg parameter
- ZKOVN a real square matrix containing either the least square matrix $(\underline{\underline{X}}^T \underline{\underline{X}})$ or the approximated second order partial derivative matrix

SWITCH POSITIONS:

IDOUB = 0 initial position of the switch

> 0 indicates that the step size has been
increased by a certain value

IIATT = 0 initial position of the switch

= 1 indicates that increasing the step size has
given estimates that are poorer than the last
estimates

SUBROUTINES CALLED:

AFILT filters an array using an autoregressive filter

BCOMP computes the matrix of second order partial derivatives of the loss function

CCOMP computes the gradient of the loss function

INVERT computes the inversion of a given matrix

PARSP prints out the current model order results

PREDS computes the deterministic predictions of the discrete model

PRINT prints out repetitive headings

RESID computes the residuals of the estimated model

3.11 SUBROUTINE: IVOFF

PURPOSE: to calculate the coefficients of a model of given order, choosing the best time delay over the allowed range, using the instrumental variable technique. The flow diagrams for the IVOFF module are figures 3-19 and 3-20.

METHOD: unbiased estimates of a least squares solution can be obtained by premultiplication of equation (2.12) by \underline{W}^T , the instrumental variable matrix. This leads to the estimate

$$\hat{\underline{\theta}} = [\underline{W}^T \underline{X}]^{-1} \underline{W}^T \underline{y} \quad (3.6)$$

where: 1) \underline{y} is the output sequence

2) \underline{W} is the instrumental matrix

3) \underline{X} is the input/output matrix

4) $\hat{\underline{\theta}}$ is the vector of estimated parameters

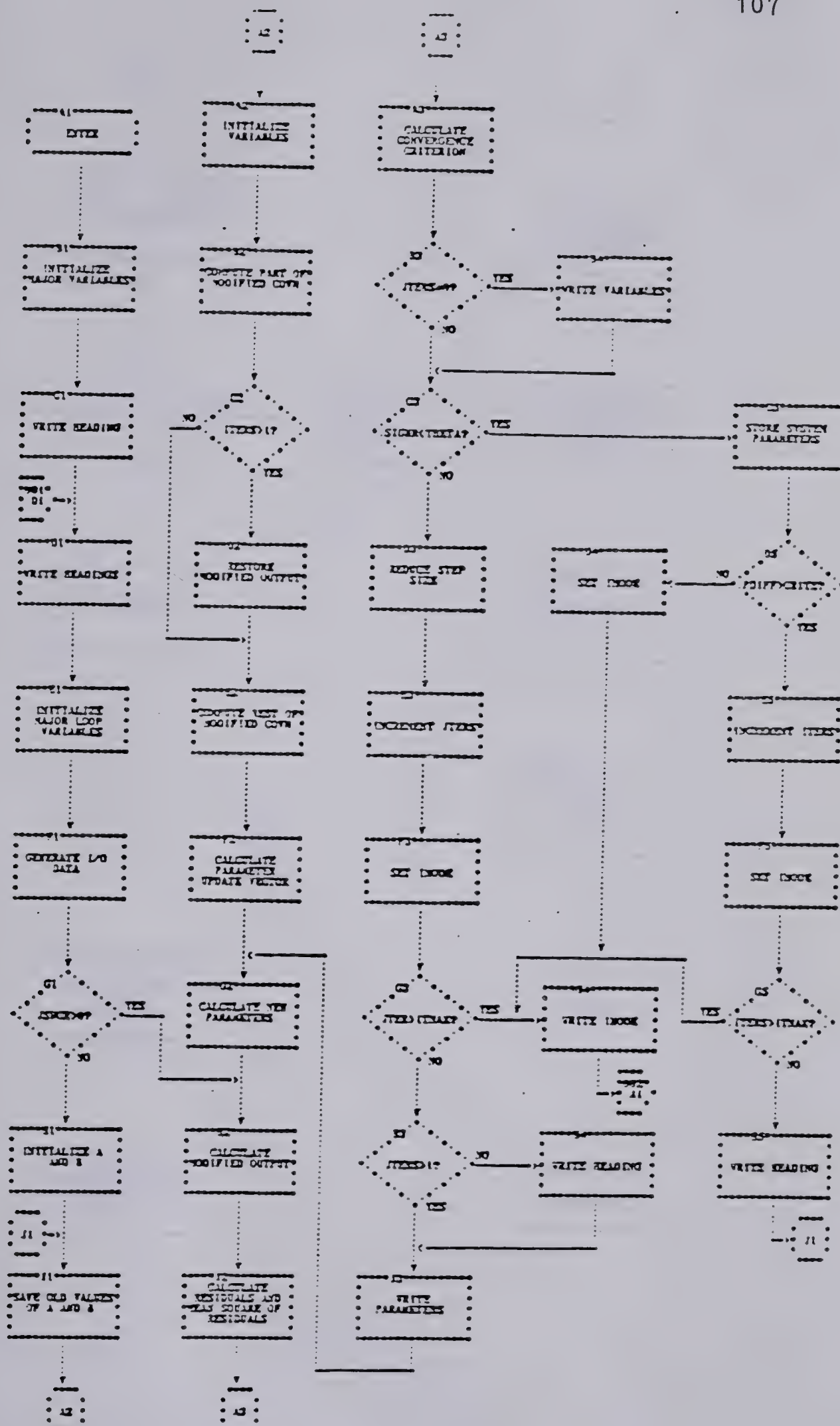


Figure 3-19 Flow Diagram for IVOFF

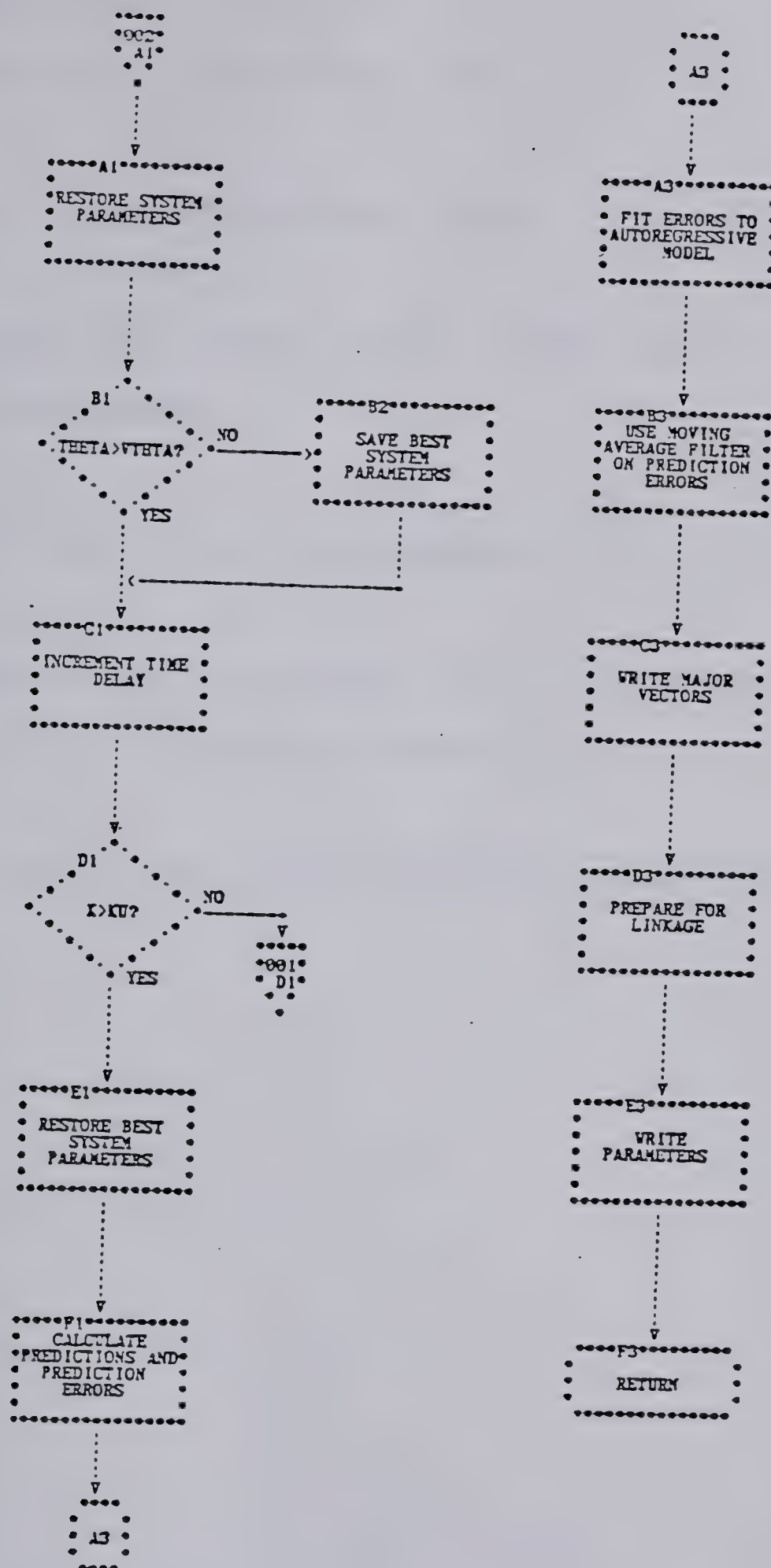


Figure 3-20 Flow Diagram for IVOFF

SUBROUTINES CALLED:

BCOMP computes the instrumental matrix

CCOMP computes the instrumental vector

PREDS computes the deterministic predictions of the
discrete model

PRINT prints out repetitive headings

RCURG computes the parameters of an autoregressive
model by a recursive procedure

RUNAV filters an array using a moving average filter

3.12 SUBROUTINE: UBIAS

PURPOSE: to calculate the coefficients of a model of given order, choosing the best time delay over the allowed range using an unbiased estimator [16] based on Slutsky's theorem. The flow diagrams for the UBIAS module are figures 3-21, 3-22 and 3-23.

METHOD: unbiased estimates of a least square solution can be obtained by subtracting the covariance matrix and vector, of the residuals between the process and model outputs, from the least squares matrix and vector calculations for the estimated parameters

$$\hat{\underline{\theta}} = [\underline{X}^T \underline{X} - \underline{R}]^{-1} [\underline{X}^T \underline{y} - \underline{Q}] \quad (3.7)$$

where: 1) \underline{y} is the output sequence

2) \underline{X} is the input/output matrix

3) $\hat{\underline{\theta}}$ is the vector of estimated parameters

4) \underline{R} is the covariance matrix of the residuals

5) \underline{Q} is the covariance vector of the residuals

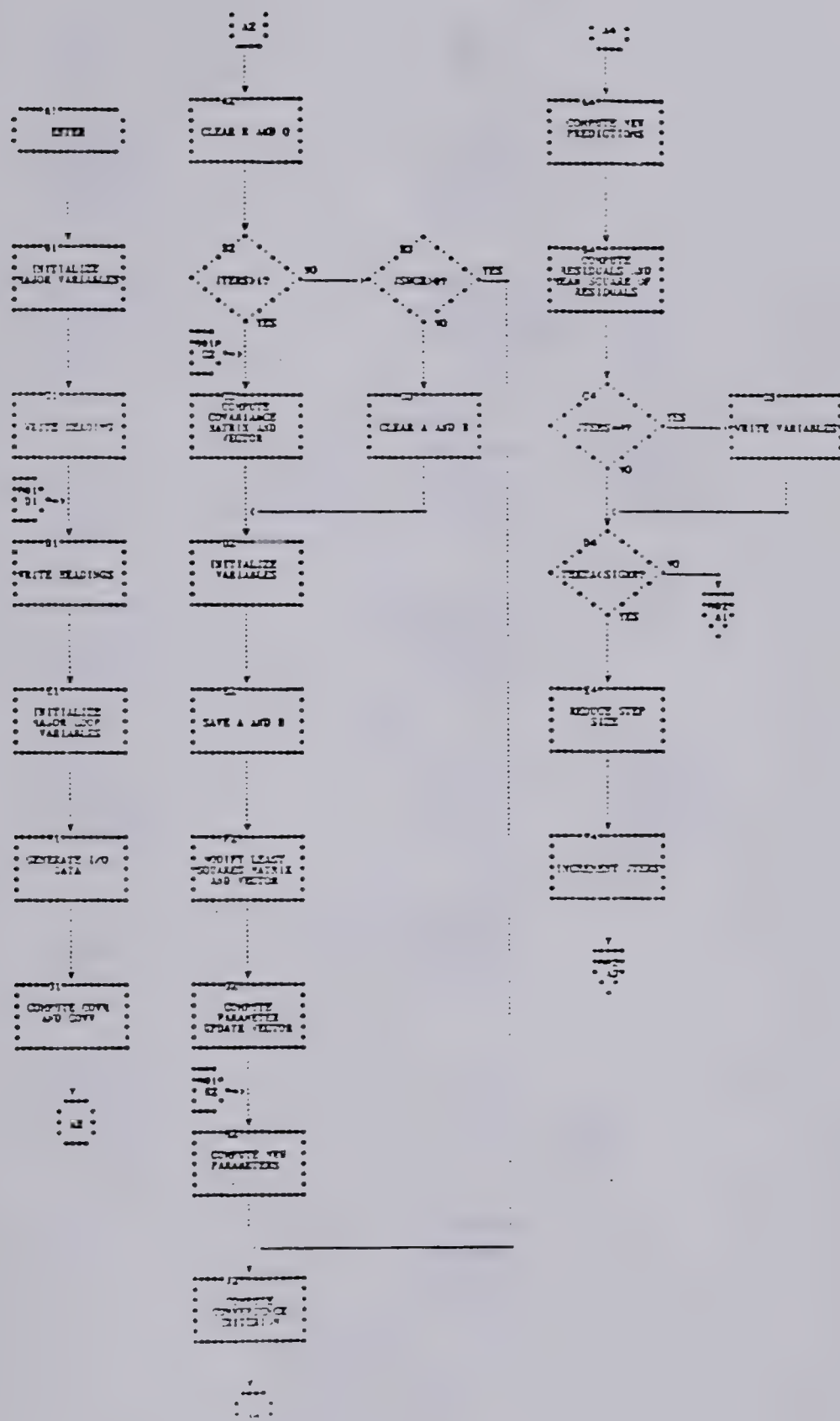


Figure 3-21 Flow Diagram for UBIAS

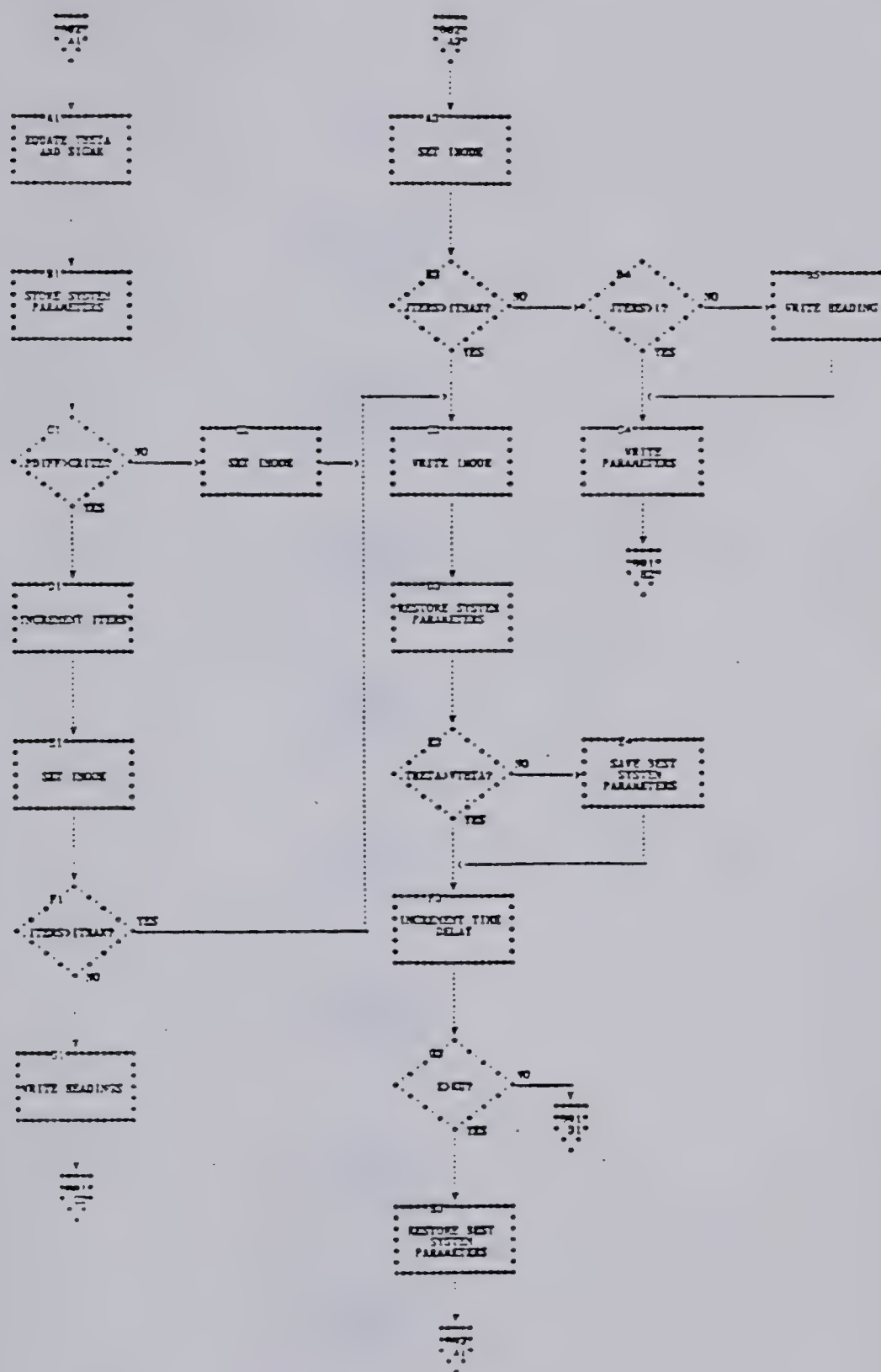


Figure 3-22 Flow Diagram for UBIAS

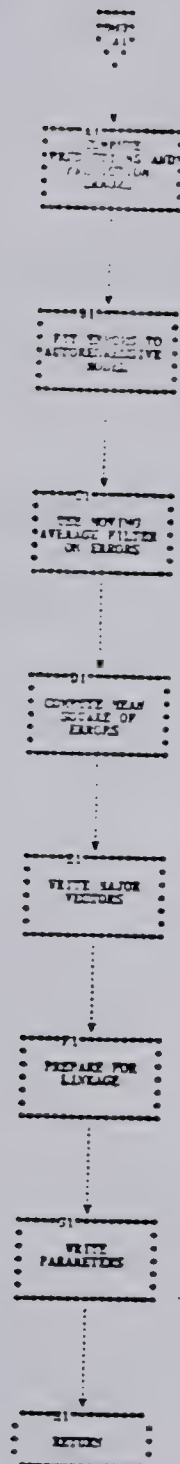


Figure 3-23 Flow Diagram for UBIAS

VARIABLES:

COM a real square matrix used to hold the modified
 COVM matrix

COV a real vector used to hold the modified COVV
 vector

Q a real vector containing the covariance vector of
 the residuals

R a real square matrix containing the covariance
 matrix of the residuals

SUBROUTINES CALLED:

BCOMP computes the covariance matrix of the residuals

CCOMP computes the covariance vector of the residuals

PREDS computes the deterministic predictions of the
discrete model

PRINT prints out repetitive headings

RCURG computes the parameters of an autoregressive
model by a recursive procedure

RUNAV filters an array using a moving average filter

3.13 SUBROUTINE: DIAGN

PURPOSE: to take a model of given order and perform various tests to determine whether the model is a satisfactory fit to the data and whether the model is a significant improvement over lower order models. The flow diagram for the DIAGN module is figure 3-24.

METHOD: the following statistical tests are used to determine if the current model is satisfactory:

- 1) F-ratio determines if the mean square error of the current model is significantly less than that of lower order models
- 2) Determinant of the covariance matrix, if zero, indicates a failure in the inversion of the covariance matrix (the model order may be too high)
- 3) Volume of the covariance ellipsoid gives an estimate of the volume, in parameter space, in which the true parameters are likely to be away from the estimated parameters

- 4) Parameter significances are used to determine if any parameters can be eliminated
- 5) Autocorrelation function of the residuals determines whether the residuals are white noise
- 6) Cross correlation function of the input and the residuals indicates whether the residuals are independent of the input. Residuals should be dependent on the disturbances only, there should be no dependence on the input if the time delay is correct.
- 7) Cross correlation function of the input and the prediction errors indicates whether the prediction errors are independent of the input. Prediction errors should be dependent on the disturbance only, there should be no dependence on the input if the model order is correct.

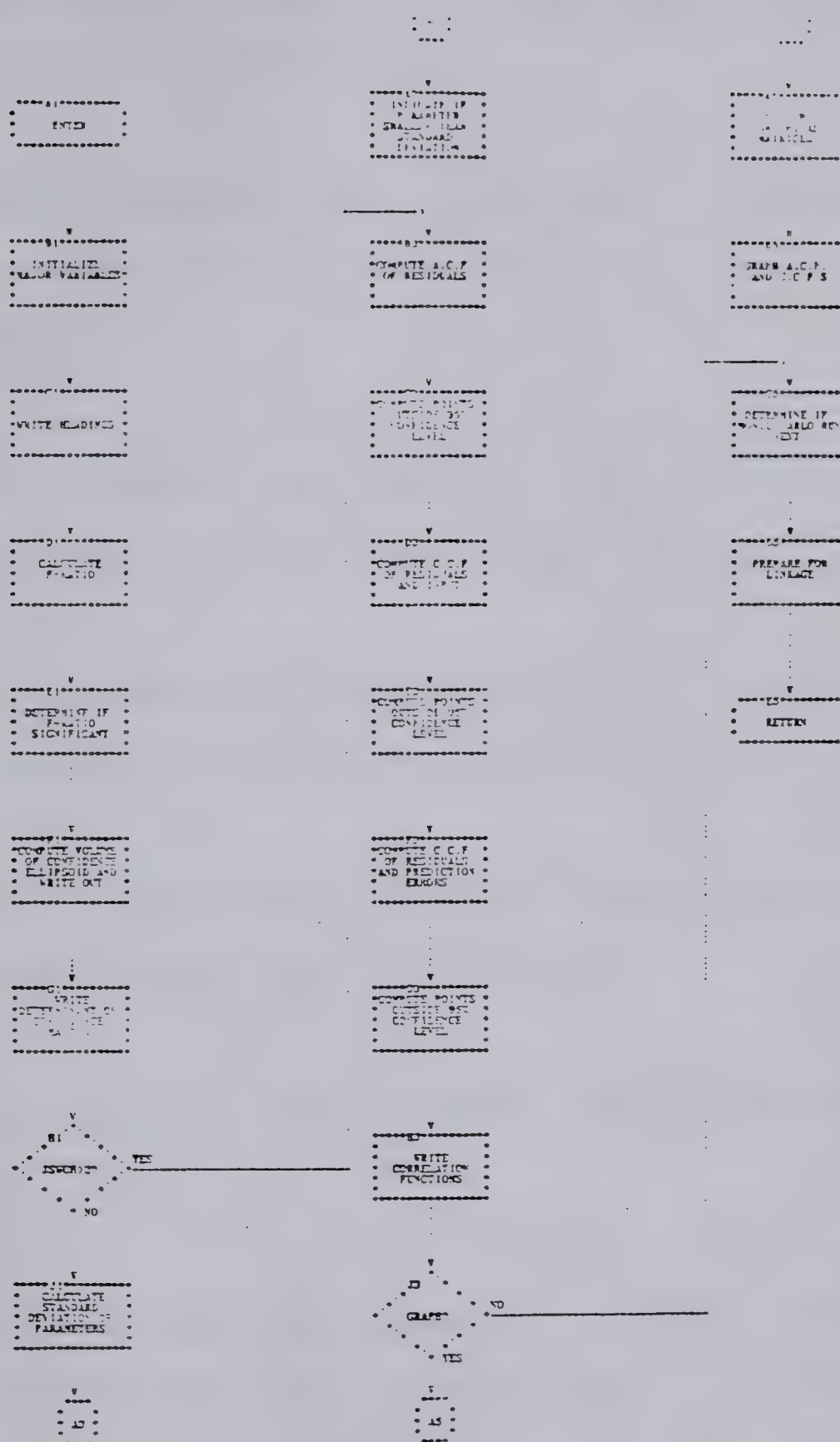


Figure 3-24 Flow Diagram for DIAGN

VARIABLES:

- ACF a real vector that contains the autocorrelation
 function of the residuals
- ACFSA a real variable that holds the normalization
 factor for the autocorrelation function of the
 residuals
- AF a real variable that contains the normalization
 factor for the cross correlation functions
- CCFUP a real vector that contains the cross correlation
 function of the input and prediction errors
- CCFUR a real vector that holds the cross correlation
 function of the input and residuals
- DIST a real variable that contains the 95% confidence
 limit
- FRAT a real variable that contains a comparison value
 for the F-ratio test to determine if the present
 model is significantly better than the previous
 model
- ISAI an integer vector that holds the position of the
 points in the cross correlation function of the

input and the prediction errors that are outside the 95% confidence limit

ISA2 an integer vector that holds the position of the points in the autocorrelation function that are outside the 95% confidence limit

ISA3 an integer vector that holds the position of the points in the cross correlation function of the input and the residuals that are outside the 95% confidence limit

JCNT an integer variable that counts the number of points in the autocorrelation function outside the 95% confidence limit

KCNT an integer variable that counts the number of points in the cross correlation function of the input and the residuals outside the 95% confidence limit

MCNT an integer variable that counts the number of points in the cross correlation function of the input and the prediction errors outside the 95% confidence limit

RATIO a real variable that contains the F-ratio value

RDET a real variable that contains the volume of the
covariance confidence ellipsoid

RR, SS real variables used for the calculations of
the standard deviation of the A and B parameters

SIGML a real variable that contains the mean square
error of the previous model order

SUBROUTINES CALLED:

DGSW supplies messages and program flow control at
decision points in the program

GRAPH allows the user to plot on a hard copy terminal,
a graph of a function supplied by the user

3.14 SUBROUTINE: MDISP

PURPOSE: to allow the user to eliminate insignificant parameters from the model and to determine the effect these eliminations have on the impulse response of the model. The flow diagram for the MDISP module is figure 3-25.

METHOD: the module has the following abilities

- 1) elimination of insignificant parameters
- 2) calculation of the model's impulse response
- 3) computation of a residual sequence using the best parameters and the modified input/output data. An autoregressive model of successive orders is fitted to this sequence. The best model is used to filter the residuals to produce an uncorrelated sequence which is an estimate of the sequence driving the system's noise model.
- 4) the best parameters are used to calculate the predictions of the output and the error sequence. An autoregressive model of successive orders is fitted to the error

sequence which is the estimated output of the system's noise model.

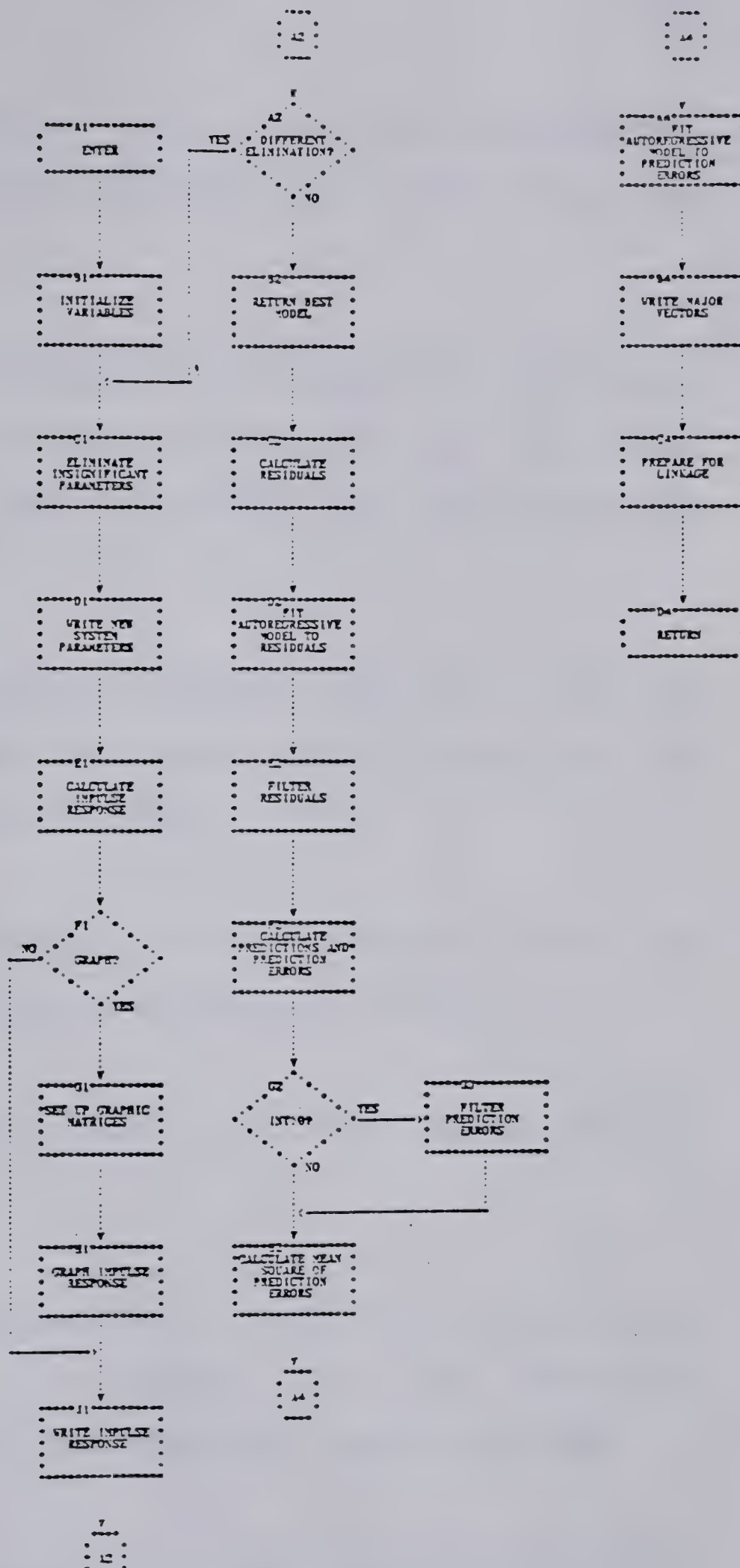


Figure 3-25 Flow Diagram for MDISP

VARIABLES:

EEM a real vector used to store intermediate calculations of the residuals and the prediction errors

INT an integer variable that contains the difference between the number of times the input and output sequences were differenced in the subroutine DMOVE {3.7}

IROMAX an integer variable that contains the maximum autoregressive model order for the minor subroutine RCURG

JFRSTM an integer variable used to store and change IFRST without destroying IFRST

OUTT a real vector used to hold the model's impulse response

SIGMM a real variable that contains the mean square values of the residuals or of the prediction errors returned by the minor subroutine RCURG

SIGS a real variable that contains the mean square value of the prediction errors for a zeroth order autoregressive model

SUBROUTINES CALLED:

DGSW supplies messages and program flow control at decision points in the program

FREAD allows the user to use free format input and provides certain error checks for the input [21]

GRAPH allows the user to plot on a hard copy terminal, a graph of a function supplied by the user

PREDS computes the deterministic predictions of the discrete model

PRINT prints out repetitive headings

PULSE calculates the impulse response of a given model

RCURG computes the parameters of an autoregressive model by a recursive procedure

RESID computes the residuals of the estimated model

RUNAV filters an array using a moving average filter

3.15 SUBROUTINE: NOISE

PURPOSE: to calculate the coefficients of a given noise model using the least squares method. The flow diagram for the NOISE module is figure 3-26.

METHOD: this program produces a noise model

$$\hat{\underline{e}} = \frac{[1+D(Z^{-1})]^{\underline{s}}}{[1+C(Z^{-1})]} \quad (3.8)$$

where: 1) $\hat{\underline{e}}$ is the prediction error sequence derived by MDISP {3.14}

2) \underline{s} is the uncorrelated driving sequence derived by MDISP {3.14}

Various model orders from 1 to the system's model order are used. The mean square error of the noise model, the noise model parameters and their standard deviations and the autocorrelation function of the noise model's residuals are calculated. The user then picks

the best noise model order and this is then passed to the subroutine FINAL {3.16}.

VARIABLES:

NN an integer variable that contains the noise model
 order to be determined

SIGMP a real variable that contains the mean square
 error of the zeroth order noise model

SUBROUTINES CALLED:

- DGSW supplies messages and program flow control at decision points in the program
- FREAD allows the user to use free format input and provides certain error checks for the input [21]
- GRAPH allows the user to plot on a hard copy terminal, a graph of a function supplied by the user
- MCOMP computes the covariance matrix, the covariance vector and the determinant of the covariance matrix
- RESID computes the residuals of the estimated model

3.16 SUBROUTINE: FINAL

PURPOSE: to display the system and noise transfer functions and to calculate the zeros and the poles of the system and the noise models. This program also allows the user to take the file display option called FDISP {3.17}. The flow diagrams for the FINAL module is figure 3-27.

METHOD: the zeros and poles of the system and the noise models are calculated by the subroutine POLTR, which is part of the IBM 360 Scientific Subroutine Package [19].

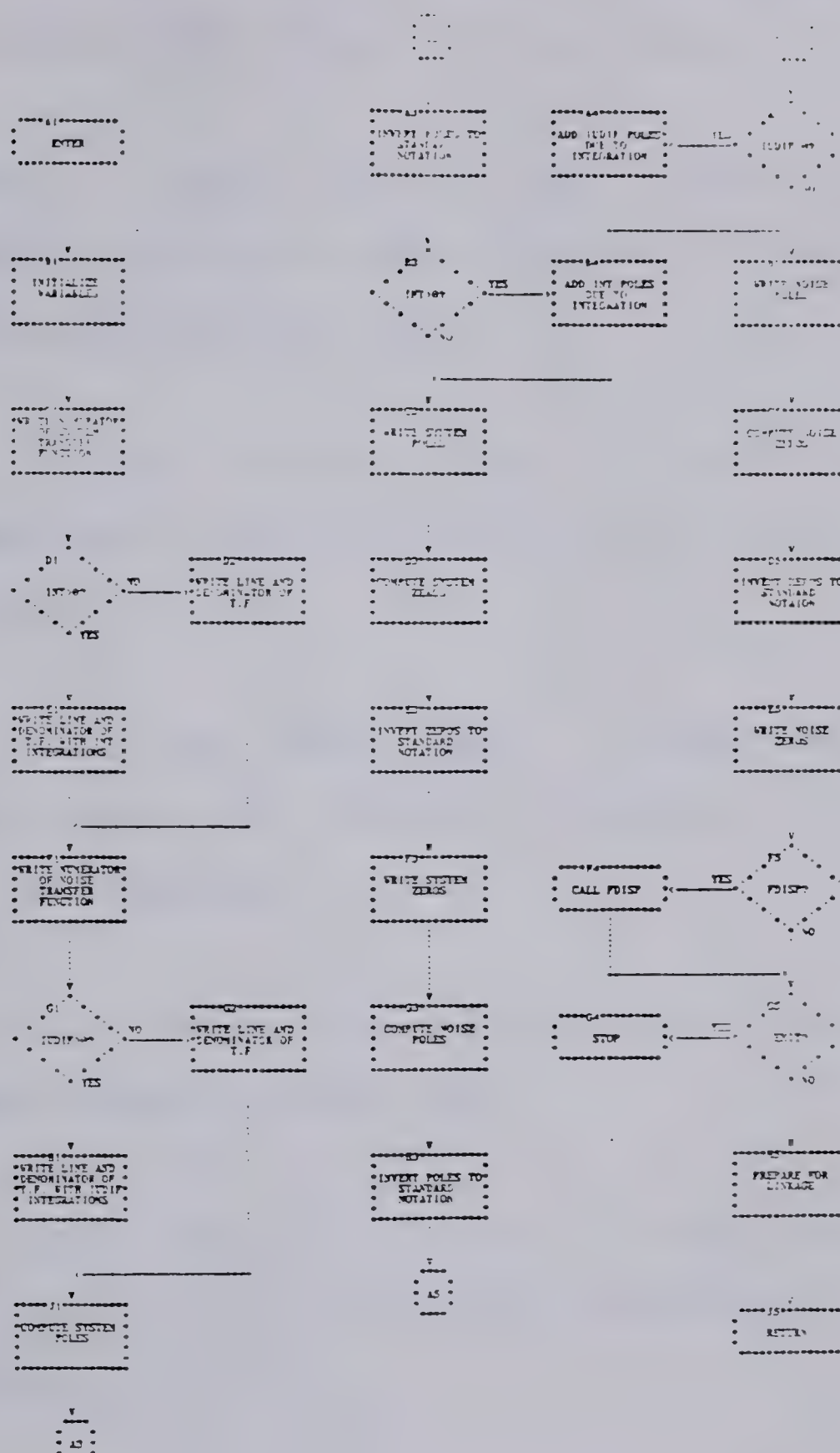


Figure 3-27 Flow Diagram for FINAL

VARIABLES:

AA, BB, CC, DD real vectors used to store the system
and the noise transfer function coefficients for
determining their poles and their zeros

BLANK, BLANKB integer vectors used in object time
formatting of the transfer function to eliminate
unnecessary numerical signs

BRAC, BRAC1 integer vectors used in object time
formatting to provide a bracket at the end of the
denominator

FMTA, FMTB, FMTC, FMTD, FMTAA, FMTCC integer vectors
that contain the original formatting of the
transfer functions

IER an integer variable that is the return error code
of the subroutine POLRT [19]

JUN1, JUN2, JUN3, JUN4, JUN11, JUN33 integer vectors
that are the work areas for the formatting of the
transfer functions

LINE an integer vector used to store the format of the
line between the numerator and the denominator

MINUS an integer vector used to store the format of the exponents of the transfer function

PLUS, PLUSB integer vectors used in object time formatting to provide a plus sign for use in the transfer functions

ROOT1 a real vector used to store the imaginary roots of the polynomial returned by the subroutine POLRT [19]

SAVE a real vector used to store the real roots of the polynomial returned by the subroutine POLRT [19]

SAVPOL a real vector used to contain the absolute value of the polynomial coefficients to be used in formatting the transfer functions

WORK a real vector used as a working area by the subroutine POLRT [19]

SUBROUTINES CALLED:

DGSW supplies messages and program flow control at
 decision points in the program

FDISP {3.17} supplies graphical displays of data files

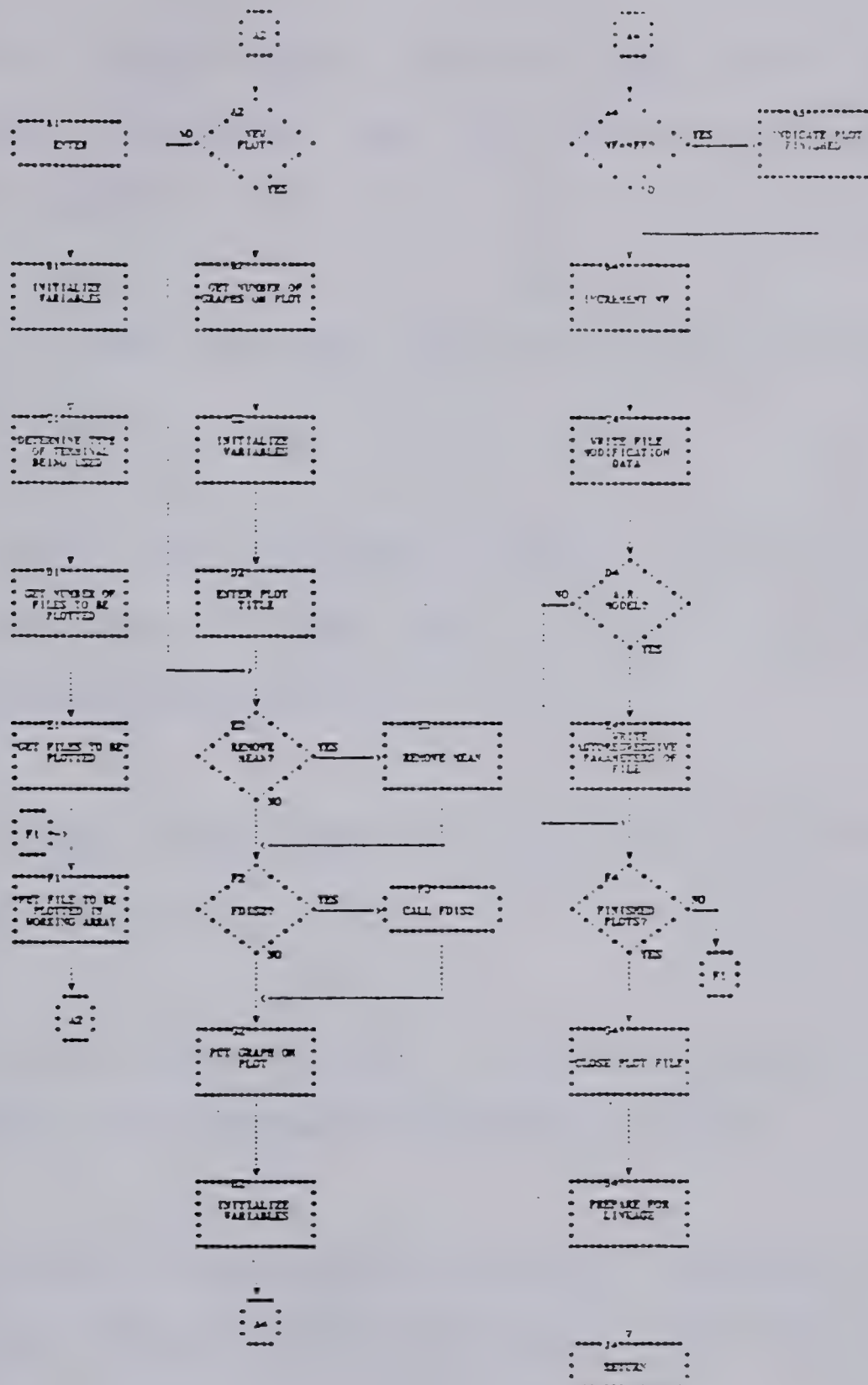
POLTR computes the real and the complex roots of a real
 polynomial [19]

3.17 SUBROUTINE: FDISP

PURPOSE: allows graphical display of data files and some file modification. Gives the user the option of major file modification by linking to the subroutine FDIS2 {3.18}. The flow diagram for the FDISP module is figure 3-28.

DATA FILE NUMBERS:

- 1 U, unmodified input
- 2 Y, unmodified output
- 3 E, errors after DAGEN {3.6}
- 4 U1, modified input
- 5 Y1, modified output
- 6 E1, residuals
- 7 Y2, predictions of the output
- 8 E2, prediction errors



VARIABLES:

- AA a real variable that contains the mean of the file being considered
- ALPH a real vector that contains the title, the abscissa designator and the ordinate designator of the plot
- BLANKS a real variable used to clear the real vector ALPH
- IDIF1 an integer variable that contains the number of differencings of the file to be done in the subroutine FDIS2 {3.18}
- IFF an integer vector used to store the file numbers to be plotted
- IFX an integer variable that is used as a switch to indicate if a particular plot is finished
- INIT an integer variable that is used to determine the type of terminal that the plotting will be done at
- IT1 an integer variable that is used to calculate the time constant (s-space) for the high pass filter

of the file to be operated on in the subroutine
FDIS2 {3.18}

KC an integer variable that indicates the type of
graph to be plotted by the subroutine CGPL [20]

KD an integer variable that is used as a dummy
variable in the subroutine CGPL [20]

NF an integer variable that contains the current
number of graphs on the plot under construction

NFF an integer variable that contains the maximum
number of graphs to be done on the plot

NO an integer variable that contains the order of
the autoregressive model of the file operated on
in the subroutine FDIS2 {3.18}

NUM an integer variable that contains the number of
files to be plotted

XCOR a real vector containing the autoregressive
parameters of the file being considered

X9, Y9, Z9 real vectors used in the subroutine CGPL
[20] to store the value of the abscissa, the
ordinate and a third parameter which is a dummy
variable in this program

SUBROUTINES CALLED:

CGPL plotting routine supplied by MTS applications
 subroutines [20]

DGSW supplies messages and program flow control at
 decision points in the program

FDIS2 {3.18} does file modifications and calculations
 on the current file

GRAPH allows the user to plot on a hard copy terminal,
 a graph of a function supplied by the user

3.18 SUBROUTINE: FDIS2

PURPOSE: to modify, to find auto and cross correlation functions and to fit an autoregressive model to the file being considered. The flow diagram for the FDIS2 module is figure 3-29.

METHOD: the program uses the following to modify the file:

- 1) high pass filtering
- 2) differencing
- 3) moving average filtering

The program also offers plots of the auto and cross correlation functions.

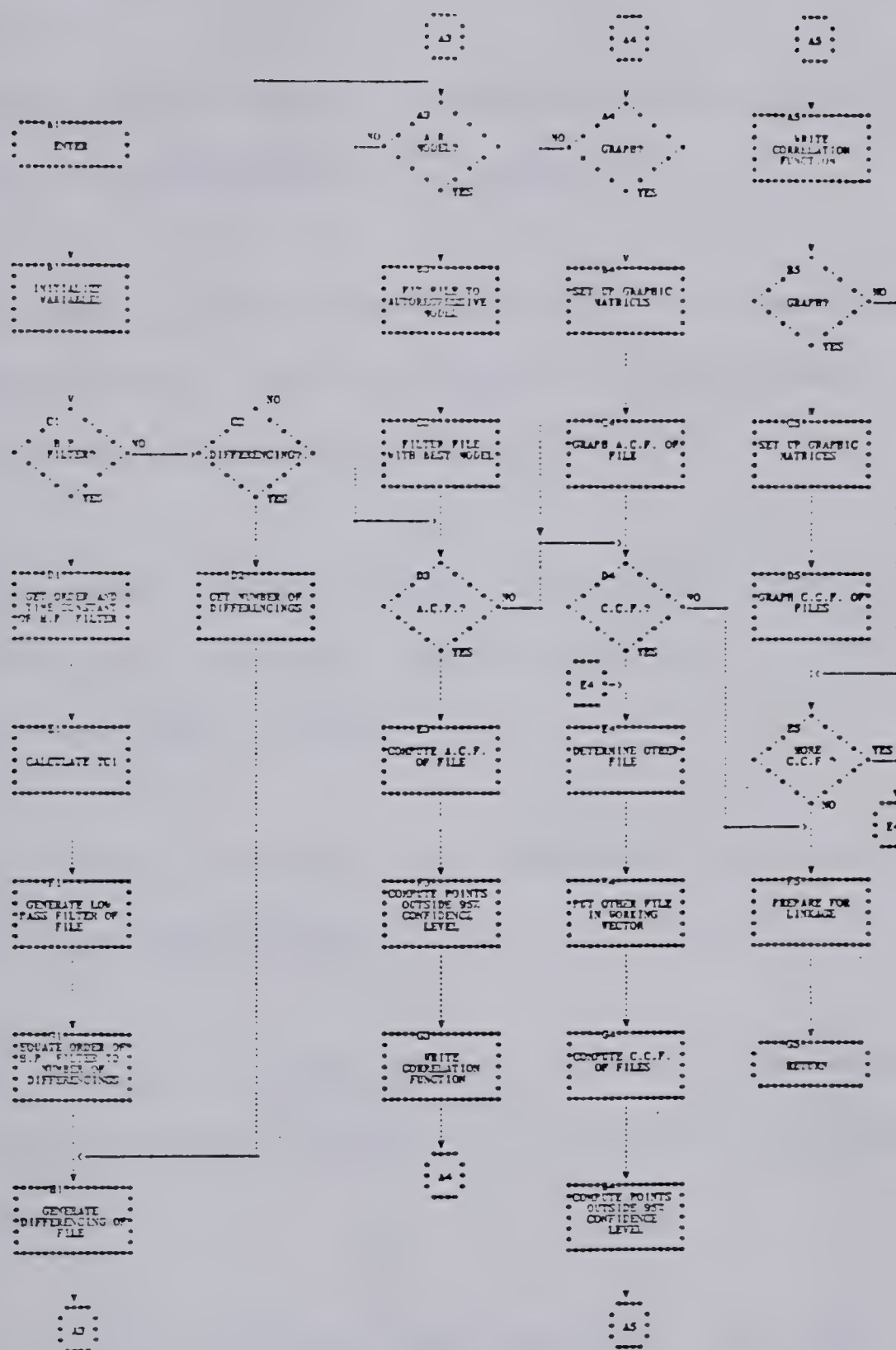


Figure 3-29 Flow Diagram for FDIS2

VARIABLES:

CCF a real vector that contains the cross correlation function

EF a real vector used to store the second file for the cross correlation function

I3 an integer vector that holds the position of the points in the cross correlation function that are outside the 95% confidence limit

MFRST an integer variable that is used to contain the number of the first data element to be used in the high pass filter

NU an integer variable that contains the order of the high pass filter

NOMAX an integer variable that contains the maximum autoregressive order for the minor subroutine RCURG

NUMF an integer variable that contains the number of the file to be used in the cross correlation function calculation

SIGMF a real variable that contains the mean square value of the file returned by the minor subroutine RCURG

TC1 a real variable that contains the time constant of the high pass filter for the file

SUBROUTINES CALLED:

DGSW supplies messages and program flow control at decision points in the program

FREAD allows the user to use free format input and provides certain error checks for the input [21]

GRAPH allows the user to plot on a hard copy terminal, a graph of a function supplied by the user

RCURG computes the parameters of an autoregressive model by a recursive procedure

RUNAV filters an array using a moving average filter

CHAPTER FOUR

PARAMETER ESTIMATION OF A SIMULATED PROCESS USING UNALIP

4.1 INTRODUCTION

This section compares the performance and the computational time of the four identification methods used in the UNiversity of ALberta Identification Package (UNALIP). All the methods are off-line identification routines that lead to parametric models. This chapter has been organized in a manner similar to the formats used by Isermann [2] and Saridis [28].

Simulated processes were used because:

- 1) the systems and disturbances were known exactly and could be duplicated
- 2) the input and the noise/signal ratio type could be varied easily

The methods of identification used were:

- 1) Generalized Least Squares (STIME) {3.6}
- 2) Maximum Likelihood (ASTRM) {3.7}
- 3) Instrumental Variables (IVOFF) {3.8}
- 4) Unbiased Estimator (UBIAS) {3.9}

4.2 TEST CASES

Three simulated, linear, time invariant processes were used as test cases. These processes were:

1) Second order oscillating process

$$\begin{aligned}
 G_1(z^{-1}) &= \frac{B(z^{-1})}{[1+A(z^{-1})]} \\
 &= \frac{1.0z^{-1} + 0.5z^{-2}}{1 - 1.5z^{-1} + 0.7z^{-2}} \quad (4.1)
 \end{aligned}$$

2) Second order nonminimum phase process

$$\begin{aligned}
 G_2(z^{-1}) &= \frac{B(z^{-1})}{[1+A(z^{-1})]} \\
 &= \frac{-0.102z^{-1} + 0.173z^{-2}}{1 - 1.425z^{-1} + 0.496z^{-2}} \quad (4.2)
 \end{aligned}$$

3) Third order low pass delay process

$$\begin{aligned} G_3(z^{-1}) &= \frac{z^{-1}B(z^{-1})}{[1+A(z^{-1})]} \\ &= \frac{0.065z^{-2} + 0.048z^{-3} - 0.008z^{-4}}{1 - 1.5z^{-1} + 0.0705z^{-2} - 0.1z^{-3}} \quad (4.3) \end{aligned}$$

Figure 4-1 is a general block diagram of the simulated processes.

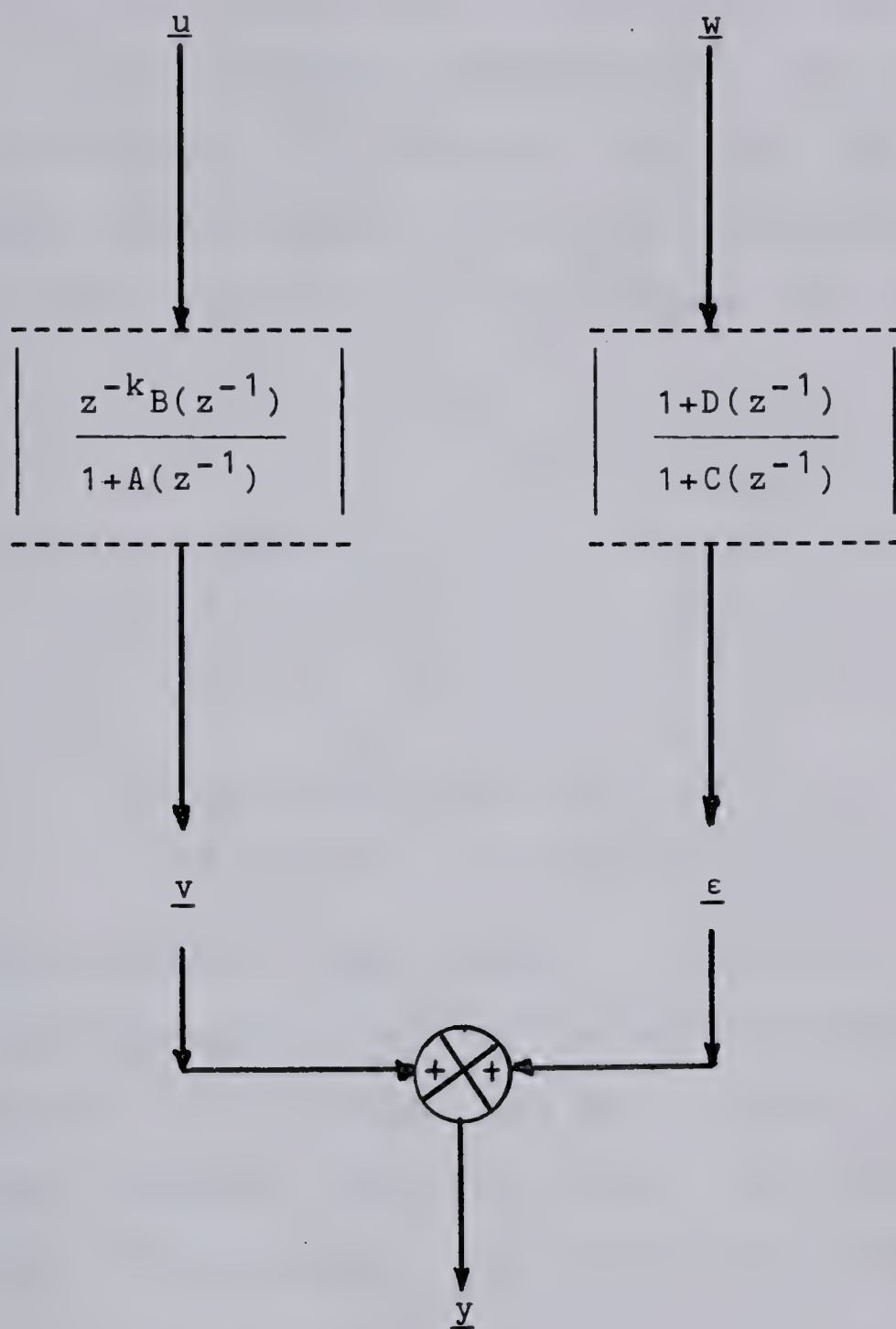


Figure 4-1 General Block Diagram of Simulated Processes

The output, \underline{y} , of the system to be identified was contaminated by an autocorrelated discrete noise sequence which was generated by a noise filter $G_n(z^{-1})$. The filter was driven by a discrete white noise sequence, \underline{w} , with normal distribution, zero mean and variable variance, σ^2 , depending on the type of the noise/signal ratio desired. The same discrete transfer function used by Eykhoff [9] was employed for all three systems:

$$G_n(z^{-1}) = \frac{1 + D(z^{-1})}{1 + C(z^{-1})}$$

$$= \frac{1}{1 - 1.0z^{-1} - 0.2z^{-2}} \quad (4.4)$$

The discrete white noise input of the noise filter $G_n(z^{-1})$ was produced by an IBM System Subroutine Package [19] routine called GAUSS which generated Gaussian distributed numbers with zero mean and variance as specified. Therefore the noise sequence was reproducible.

4.3 IDENTIFICATION

The input signal to all processes, a pseudo random binary sequence, was generated in DAGEN {3.6} by accessing the minor subroutine PRBS. The pseudo random binary sequence, the raw input, had a length of 511 and the two states produced were -1.0 and 1.0. Figure 4-2 is a plot of the raw input sequence used in all runs. DAGEN {3.6} took the raw input sequence and passed it through a specified process model which produced a raw output sequence. The noise sequence was generated and added to the raw output sequence. Filtering of the raw input or output was not utilized in this study. The raw input and output values were passed to DMOVE {3.7} where the values were transformed into mean free modified input and output sequences. No filtering of the modified input or output was employed so the modified input and output values were used by the estimation routines to determine the process model.

Tests were performed for two noise/signal ratio types at input/output sequence lengths of 100, 300, and 500 data pairs. The error of the estimated parameters and the error of the resulting impulse response for the estimated systems were tabulated for all runs. The errors were calculated from formulas provided by Isermann [2]:

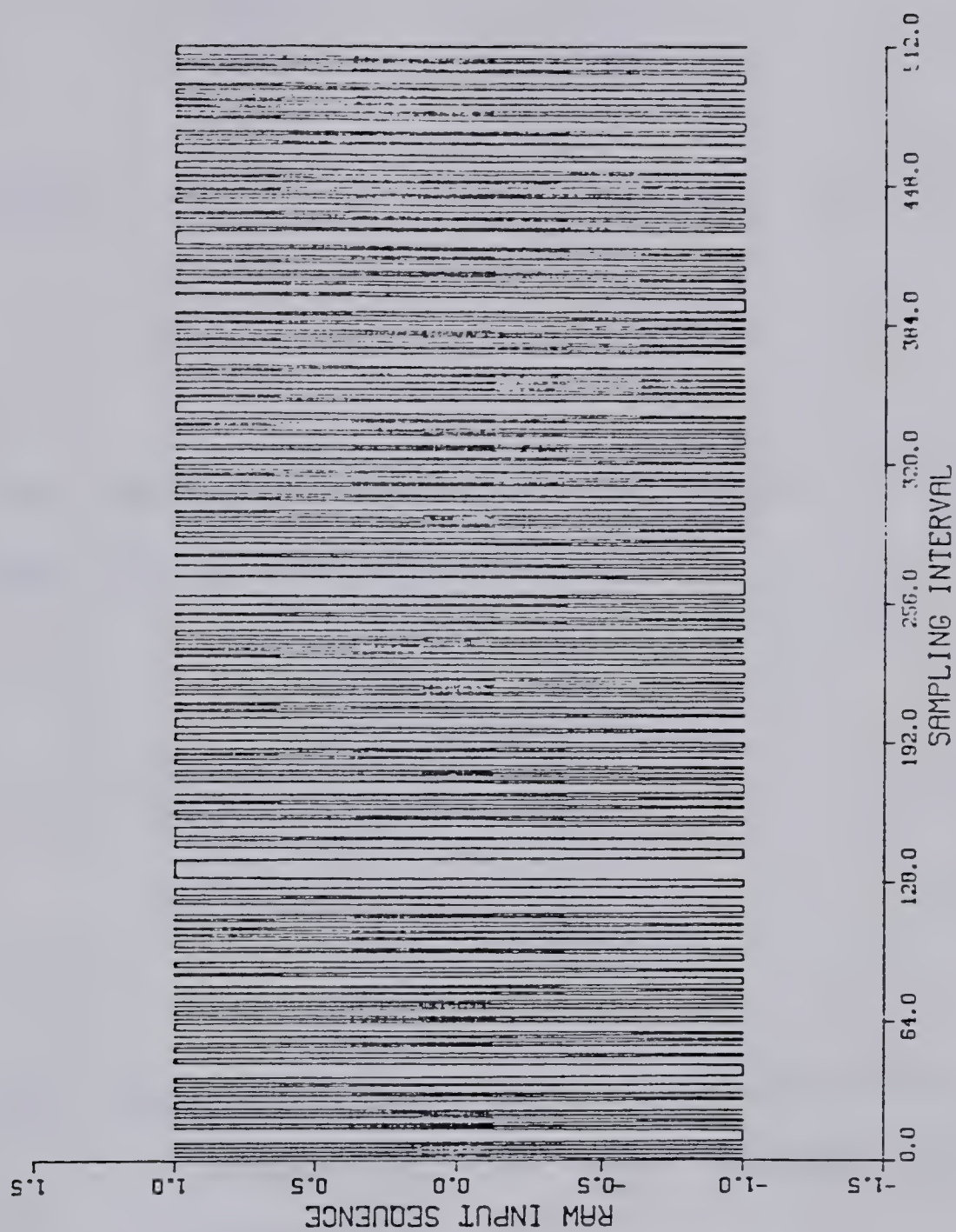


Figure 4-2 Pseudo Random Binary Sequence Used as Input for Simulation Runs

a) Squared relative parameter errors

$$\delta_{\Sigma 1} = \left\| \left\| \frac{\Delta \theta_i}{\theta_i} \right\| \right\| = \left[\sum_{i=1}^{2n} \left(\frac{\Delta \theta_i}{\theta_i} \right)^2 \right]^{1/2} \quad (4.5)$$

where: 1) $\delta_{\Sigma 1}$ is the squared relative parameter error

$$2) \Delta \theta_i = \hat{\theta}_i - \theta_i$$

b) Mean squared parameter errors related to the mean squared true parameters

$$\delta_{\Sigma 2} = \frac{\left\| \left\| \Delta \theta_i \right\| \right\|}{\left\| \left\| \theta_i \right\| \right\|} = \left[\frac{\sum_{i=1}^{2n} \Delta \theta_i^2}{\sum_{i=1}^{2n} \theta_i^2} \right]^{1/2} \quad (4.6)$$

where $\delta_{\Sigma 2}$ is the mean square parameter error related to the mean squared true parameters

c) Mean squared impulse response errors related to the mean squared true impulse function values

$$\delta_g = \left[\frac{\Delta g_t^2}{g_t^2} \right]^{1/2} = \left[\frac{T_s \sum_{t=0}^{\infty} \Delta g_t^2}{T_s \sum_{t=0}^{\infty} g_t^2} \right]^{1/2} \quad (4.7)$$

where: 1) $\Delta g_t = \hat{g}_t - g_t$

2) g_t is the impulse response sequence of the true system

3) \hat{g}_t is the impulse response sequence of the estimated system

4) T_s is the settling time of the system

5) δ_g is the mean squared impulse response error related to the mean squared true impulse function values

To obtain a better comparison of the four identification methods, five runs with different noise sequences were used for each noise/signal ratio type. The noise/signal ratio types used were low and high. Low or high noise/signal ratios types are generated by manipulating the standard deviations of the Gaussian distribution driving the noise filter. Noise/signal

ratios are calculated using the formulation originally encoded in OXIP:

$$\eta = \left[\left(\frac{\sum_{i=1}^N \epsilon_i}{\sum_{i=1}^N v_i} \right) \right]^{1/2} \quad (4.8)$$

The noise sequences were generated using a different seed for the subroutine GAUSS [19]. The seeds used were 1, 3, 5, 7 and 9. The noise/signal ratio values for the five different noise sequences and the three different processes are given in Tables 4.1 and 4.2. The standard deviation values for the low and high noise/signal ratios were 0.1 and 5.0 for process I and 0.01 and 0.1 for processes II and III. The reduction of the standard deviation values for processes II and III was due to the high noise/signal ratios generated using the standard deviation values 0.1 and 5.0 for these processes. The standard deviation values were reduced until the noise/signal ratios were similar for all three processes.

To obtain results that could be compared to Isermann's [2], the standard deviations of the errors were calculated using the following relationships for the five noise sequences

$$\sigma_{\delta_{\Sigma 1}} = \left[\left(\frac{\sum_{i=1}^5 \delta_{\Sigma 1 i}^2}{4} \right) \right]^{1/2} \quad (4.9)$$

TABLE 4.1: High Noise/Signal Ratios for Each Noise Sequence and Process

process	1	3	5	7	9
I	1.940	1.984	2.195	2.088	2.242
II	5.454x10 ⁻¹	5.577x10 ⁻¹	6.169x10 ⁻¹	5.870x10 ⁻¹	6.302x10 ⁻¹
III	5.097x10	5.210x10	5.767x10	5.479x10	5.893x10

TABLE 4.2: Low Noise/Signal Ratios for Each Noise Sequence and Process

process	1	3	5	7	9
I	3.880x10 ⁻²	3.968x10 ⁻²	4.389x10 ⁻²	4.176x10 ⁻²	4.484x10 ⁻²
II	5.454x10 ⁻²	5.577x10 ⁻²	6.169x10 ⁻²	5.870x10 ⁻²	6.302x10 ⁻²
III	5.097x10	5.210x10	5.767x10	5.479x10	5.893x10

where $\sigma_{\delta_{\Sigma 1}}$ is the standard deviation of $\delta_{\Sigma 1}$,

$$\sigma_{\delta_{\Sigma 2}} = \left[\left(\sum_{i=1}^5 \delta_{\Sigma 2_i}^2 \right) / 4 \right]^{1/2} \quad (4.10)$$

where $\sigma_{\delta_{\Sigma 2}}$ is the standard deviation of $\delta_{\Sigma 2}$, and

$$\sigma_{\delta_g} = \left[\left(\sum_{i=1}^5 \delta_{g_i}^2 \right) / 4 \right]^{1/2} \quad (4.11)$$

where σ_{δ_g} is the standard deviation of δ_g .

The average standard deviations of the parameters were also calculated and used as error measures using the expression

$$\sigma_{\theta_i} = \left[\sum_{j=1}^5 (\hat{\theta}_i - \theta_{ij})^2 / 4 \right]^{1/2} \quad (4.12)$$

where: 1) σ_{θ_i} is the standard deviation of the "i" th parameter

2) θ_i is the "i" th system parameter

3) $\hat{\theta}_i$ is the estimate of the "i" th system parameter

Each of the parameter identification methods, STIME {3.6}, ASTRM {3.7}, IVOFF {3.8} and UBIAS {3.9} have been applied to the three processes using:

- 1) two noise/signal ratios
- 2) five different noise sequences
- 3) three different input/output sequence lengths

Therefore, $4 \times 3 \times 2 \times 5 \times 3 = 360$ identification runs were used to compile the results. Tables 4.3 to 4.6 are tabulations of the standard deviations of the error measures for the four estimation routines. Tables 4.7 to 4.10 are tabulations of the average standard deviations of the process parameters for the four estimation routines.

The computational time of the identification routines was roughly estimated by determining the amount of CPU time (seconds) required for an identification method to estimate the parameters of a particular system with a fixed number of data pairs. This estimation is similar to the comparison done by Saridis [28]. Two cases were used to obtain data for computational expense for the four estimation routines

- 1) low noise/signal ratio and an input of 100 data pairs
- 2) high noise/signal ratio and an input of 500 data pairs

The input data was the same as the other identification runs. The seed of the noise generator was 1 for all

TABLE 4.3 Comparison of the Standard Deviations of the Error Measures for Generalized Least Squares (STIME)

STIME		low						high		
noise/signal		100	300	500		100	300	500		
run length										
process I	σ_{δ}^g	1.389×10^{-2}	4.662×10^{-3}	3.002×10^{-3}		0.569	0.411	0.361		
	$\sigma_{\delta \Sigma 1}$	4.762×10^{-2}	2.257×10^{-2}	1.318×10^{-2}		2.198	1.489	1.098		
	$\sigma_{\delta \Sigma 2}$	1.319×10^{-2}	6.215×10^{-3}	3.610×10^{-3}		0.669	0.443	0.339		
process II	σ_{δ}^g	2.382×10^{-2}	1.350×10^{-2}	8.481×10^{-3}		1.451	0.358	0.273		
	$\sigma_{\delta \Sigma 1}$	1.695×10^{-2}	9.581×10^{-3}	5.478×10^{-3}		0.266	0.240	0.161		
	$\sigma_{\delta \Sigma 2}$	5.141×10^{-3}	3.089×10^{-3}	1.787×10^{-3}		0.119	0.110	7.462×10^{-2}		
process III	σ_{δ}^g	1.993×10^{-2}	1.447×10^{-2}	1.777×10^{-2}		9.658×10^{-2}	7.799×10^{-2}	5.107×10^{-2}		
	$\sigma_{\delta \Sigma 1}$	2.928	4.557	4.386		5.378	5.573	5.278		
	$\sigma_{\delta \Sigma 2}$	0.338	0.519	0.501		0.524	0.536	0.539		

TABLE 4.4: Comparison of the Standard Deviations of the Error Measures for Maximum Likelihood (ASTRM)

ASTRM											
noise/signal		low						high			
run length		100	300	500		100	300	500			
process I	σ_{δ}^g	1.637×10^{-2}	5.114×10^{-3}	3.312×10^{-3}		0.653	0.429	0.362			
	$\sigma_{\delta \Sigma 1}$	5.173×10^{-2}	2.053×10^{-2}	1.213×10^{-2}		1.944	1.238	0.819			
	$\sigma_{\delta \Sigma 2}$	1.401×10^{-2}	5.881×10^{-3}	3.454×10^{-3}		0.556	0.359	0.240			
process II	σ_{δ}^g	1.951×10^{-2}	1.005×10^{-2}	6.086×10^{-3}		0.188	0.106	6.578×10^{-2}			
	$\sigma_{\delta \Sigma 1}$	1.645×10^{-2}	8.971×10^{-3}	5.159×10^{-3}		0.167	8.759×10^{-2}	5.136×10^{-2}			
	$\sigma_{\delta \Sigma 2}$	4.231×10^{-3}	2.435×10^{-3}	1.409×10^{-3}		8.612×10^{-2}	2.247×10^{-2}	1.383×10^{-2}			
process III	σ_{δ}^g	3.223×10^{-2}	1.147×10^{-2}	9.660×10^{-3}		0.196	8.000×10^{-2}	4.974×10^{-2}			
	$\sigma_{\delta \Sigma 1}$	3.795	2.972	1.964		5.434	5.017	4.137			
	$\sigma_{\delta \Sigma 2}$	0.451	0.341	0.244		0.664	0.598	0.479			

TABLE 4.5: Comparison of the Standard Deviations of the Error Measures for Instrumental Variables (IVOFF)

IVOFF									
noise/signal	run length	low			high				
		100	300	500	100	300	500		
process I	σ_{δ}^g	1.339×10^{-2}	5.167×10^{-3}	3.320×10^{-3}	0.463	0.249	0.160		
	$\sigma_{\delta \Sigma 1}$	7.516×10^{-2}	2.145×10^{-2}	1.250×10^{-2}	2.141	1.094	0.637		
	$\sigma_{\delta \Sigma 2}$	1.286×10^{-2}	5.932×10^{-3}	3.440×10^{-3}	0.588	0.298	0.173		
process II	σ_{δ}^g	2.414×10^{-2}	1.016×10^{-2}	6.149×10^{-3}	0.240	0.111	6.846×10^{-2}		
	$\sigma_{\delta \Sigma 1}$	1.723×10^{-2}	8.852×10^{-3}	5.055×10^{-3}	0.187	8.800×10^{-2}	5.039×10^{-2}		
	$\sigma_{\delta \Sigma 2}$	4.985×10^{-3}	2.351×10^{-3}	1.331×10^{-3}	5.666×10^{-2}	2.348×10^{-2}	1.347×10^{-2}		
process III	σ_{δ}^g	2.229×10^{-2}	1.046×10^{-2}	1.679×10^{-2}	0.156	8.794×10^{-2}	5.098×10^{-2}		
	$\sigma_{\delta \Sigma 1}$	4.331	3.831	4.288	5.347	4.197	3.498		
	$\sigma_{\delta \Sigma 2}$	0.509	0.436	0.486	0.580	0.485	0.420		

TABLE 4.6: Comparison of the Standard Deviations of the Error Measures for Unbiased Estimator (UBIAS)

UBIAS									
noise/signal	run length	low			high				
		100	300	500	100	300	500		
process I	σ_{δ_g}	1.342×10^{-2}	5.205×10^{-3}	3.3877×10^{-3}	0.427	0.261	0.168		
	$\sigma_{\delta_{\Sigma 1}}$	4.631×10^{-2}	2.136×10^{-2}	1.247×10^{-2}	2.113	1.115	0.652		
	$\sigma_{\delta_{\Sigma 2}}$	1.286×10^{-2}	5.912×10^{-3}	3.434×10^{-3}	0.581	0.303	0.177		
process II	σ_{δ_g}	2.396×10^{-2}	1.020×10^{-2}	6.174×10^{-3}	0.244	0.113	6.884×10^{-2}		
	$\sigma_{\delta_{\Sigma 1}}$	1.726×10^{-2}	8.880×10^{-3}	5.057×10^{-3}	0.190	8.899×10^{-2}	5.100×10^{-2}		
	$\sigma_{\delta_{\Sigma 2}}$	5.008×10^{-3}	2.374×10^{-3}	1.335×10^{-3}	5.894×10^{-2}	2.428×10^{-2}	1.393×10^{-2}		
process III	σ_{δ_g}	2.311×10^{-2}	1.447×10^{-2}	1.777×10^{-2}	0.140	8.157×10^{-2}	5.092×10^{-2}		
	$\sigma_{\delta_{\Sigma 1}}$	4.961	4.557	4.386	4.899	5.244	5.452		
	$\sigma_{\delta_{\Sigma 2}}$	0.584	0.519	0.501	0.693	0.607	0.621		

TABLE 4.7: Comparison of the Average Standard Deviations of the Parameters Estimated by Generalized Least Squares (STIME)

STIME		low				high			
noise/signal		100	300	500		100	300	500	
process I	σ_{a1}	6.427×10^{-3}	1.782×10^{-3}	1.042×10^{-3}		0.457	0.343		0.306
	σ_{a2}	5.792×10^{-3}	1.588×10^{-3}	9.688×10^{-4}		0.444	0.316		0.285
	σ_{b1}	9.972×10^{-2}	5.496×10^{-2}	3.083×10^{-3}		0.630	0.318		0.211
	σ_{b2}	2.281×10^{-2}	1.087×10^{-2}	6.363×10^{-3}		0.993	0.682		0.488
process II	σ_{a1}	5.687×10^{-3}	9.769×10^{-3}	2.192×10^{-3}		0.150	0.134		9.151×10^{-2}
	σ_{a2}	5.024×10^{-3}	2.715×10^{-3}	1.529×10^{-3}		9.974×10^{-2}	9.861×10^{-2}		6.611×10^{-3}
	σ_{b1}	1.100×10^{-3}	6.862×10^{-4}	3.929×10^{-4}		0.414	6.853×10^{-2}		3.936×10^{-3}
	σ_{b2}	1.249×10^{-3}	5.387×10^{-4}	3.144×10^{-4}		1.437×10^{-2}	1.158×10^{-2}		8.182×10^{-3}
process III	σ_{a1}	0.323	0.502	0.485		0.523	0.563		0.553
	σ_{a2}	0.433	0.663	0.640		0.659	0.661		0.673
	σ_{a3}	0.152	0.227	0.219		0.214	0.188		0.201
	σ_{b1}	1.013×10^{-3}	6.818×10^{-2}	3.844×10^{-2}		1.046×10^{-2}	6.760×10^{-2}		3.881×10^{-2}
	σ_{b2}	2.172×10^{-2}	3.305×10^{-2}	3.165×10^{-2}		4.078×10^{-2}	4.021×10^{-2}		3.799×10^{-2}
	σ_{b3}	1.901×10^{-2}	3.011×10^{-2}	0.289		3.802×10^{-2}	4.063×10^{-2}		3.766×10^{-2}

TABLE 4.8: Comparison of the Average Standard Deviations of the Parameters Estimated by Maximum Likelihood (ASTRM)

ASTRM		low				high			
noise/signal		100	300	500		100	300	500	
process I	run length								
	σ_{a1}	6.769x10 ⁻³	1.141x10 ⁻³	7.041x10 ⁻⁴		0.170	0.188	0.152	
	σ_{a2}	6.023x10 ⁻³	1.210x10 ⁻³	7.840x10 ⁻³		0.212	0.228	0.188	
	σ_{b1}	8.620x10 ⁻²	6.403x10 ⁻³	3.661x10 ⁻³		0.568	0.315	0.181	
process II	σ_{b2}	2.504x10 ⁻²	9.707x10 ⁻³	5.752x10 ⁻³		0.915	0.573	0.373	
	σ_{a1}	4.480x10 ⁻³	2.784x10 ⁻³	1.621x10 ⁻³		4.210x10 ⁻²	2.540x10 ⁻²	1.556x10 ⁻²	
	σ_{a2}	4.527x10 ⁻³	2.276x10 ⁻³	1.304x10 ⁻³		4.370x10 ⁻²	2.109x10 ⁻²	1.323x10 ⁻²	
	σ_{b1}	1.067x10 ⁻³	6.748x10 ⁻⁴	3.880x10 ⁻³		1.137x10 ⁻²	6.772x10 ⁻³	3.862x10 ⁻³	
process III	σ_{b2}	1.428x10 ⁻³	5.919x10 ⁻³	3.433x10 ⁻³		1.429x10 ⁻²	5.839x10 ⁻³	3.345x10 ⁻³	
	σ_{a1}	0.432	0.329	0.160		0.580	0.543	0.460	
	σ_{a2}	0.578	0.436	0.212		0.862	0.721	0.611	
	σ_{a3}	0.201	0.150	7.233x10 ⁻⁴		0.369	0.261	0.217	
	σ_{b1}	1.325x10 ⁻²	7.075x10 ⁻²	4.069x10 ⁻²		1.341x10 ⁻²	6.908x10 ⁻²	3.900x10 ⁻²	
	σ_{b2}	2.390x10 ⁻²	2.176x10 ⁻²	1.039x10 ⁻³		4.876x10 ⁻²	4.079x10 ⁻²	3.294x10 ⁻²	
	σ_{b3}	2.431x10 ⁻²	1.952x10 ⁻²	9.577x10 ⁻³		2.905x10 ⁻²	3.244x10 ⁻²	2.665x10 ⁻²	

TABLE 4.9: Comparison of the Average Standard Deviations of the Parameters Estimated by Instrumental Variables (IVOFF)

IVOFF		low				high			
noise/signal		100	300	500		100	300	500	
process I	σ_{a1}	6.255×10^{-3}	1.414×10^{-3}	8.285×10^{-4}		0.182	6.146×10^{-2}	3.733×10^{-2}	
	σ_{a2}	5.373×10^{-3}	1.395×10^{-3}	8.876×10^{-4}		0.170	6.494×10^{-2}	4.276×10^{-2}	
	σ_{b1}	9.999×10^{-2}	5.493×10^{-2}	3.082×10^{-2}		0.503	0.257	0.147	
	σ_{b2}	2.217×10^{-2}	1.131×10^{-2}	6.019×10^{-3}		1.032	0.529	0.308	
process II	σ_{a1}	5.415×10^{-3}	2.724×10^{-3}	1.552×10^{-3}		5.610×10^{-3}	2.698×10^{-2}	1.566×10^{-2}	
	σ_{a2}	5.013×10^{-3}	2.144×10^{-3}	1.199×10^{-3}		6.312×10^{-2}	2.171×10^{-3}	1.221×10^{-3}	
	σ_{b1}	1.099×10^{-3}	6.861×10^{-4}	3.927×10^{-4}		1.051×10^{-2}	6.765×10^{-3}	3.899×10^{-3}	
	σ_{b2}	1.384×10^{-3}	5.680×10^{-4}	3.328×10^{-4}		1.400×10^{-2}	5.660×10^{-3}	3.287×10^{-3}	
process III	σ_{a1}	0.489	0.422	0.470		0.514	0.459	0.398	
	σ_{a2}	0.651	0.556	0.620		0.752	0.619	0.538	
	σ_{a3}	0.226	0.189	0.213		0.312	0.231	0.197	
	σ_{b1}	1.073×10^{-3}	6.253×10^{-4}	3.842×10^{-4}		1.087×10^{-2}	7.063×10^{-2}	3.918×10^{-2}	
	σ_{b2}	3.294×10^{-2}	2.790×10^{-2}	3.075×10^{-2}		4.147×10^{-2}	3.541×10^{-2}	2.848×10^{-2}	
	σ_{b3}	2.795×10^{-2}	2.537×10^{-2}	2.805×10^{-2}		3.288×10^{-2}	2.638×10^{-2}	2.171×10^{-2}	

TABLE 4.10 Comparison of the Average Standard Deviations of the Parameters Estimated by Unbiased Estimator (UBIAS)

UBIAS		low				high			
noise/signal		100	300	500		100	300	500	
run length									
process I	σ_{a1}	6.270x10 ⁻³	1.400x10 ⁻³	8.407x10 ⁻⁴		0.186	7.372x10 ⁻²	4.736x10 ⁻²	
	σ_{a2}	5.393x10 ⁻³	1.401x10 ⁻³	9.069x10 ⁻³		0.181	7.622x10 ⁻²	5.129x10 ⁻²	
	σ_{b1}	9.997x10 ⁻²	5.492x10 ⁻²	3.081x10 ⁻²		0.494	0.255	0.146	
	σ_{b2}	2.218x10 ⁻²	1.026x19 ⁻²	6.000x10 ⁻²		1.017	0.539	0.315	
process II	σ_{a1}	5.434x10 ⁻³	2.745x10 ⁻³	1.559x10 ⁻³		5.867x10 ⁻²	2.758x10 ⁻²	1.604x10 ⁻²	
	σ_{a2}	5.046x10 ⁻³	2.175x10 ⁻³	1.197x10 ⁻⁴		6.559x10 ⁻²	2.288x10 ⁻³	1.294x10 ⁻³	
	σ_{b1}	1.099x10 ⁻³	6.862x10 ⁻⁴	3.927x10 ⁻⁴		1.050x10 ⁻²	6.757x10 ⁻³	3.884x10 ⁻³	
	σ_{b2}	1.382x10 ⁻³	5.651x10 ⁻⁴	3.332x10 ⁻⁴		1.393x10 ⁻²	5.542x10 ⁻³	3.205x10 ⁻³	
process III	σ_{a1}	0.555	0.502	0.485		0.617	0.579	0.598	
	σ_{a2}	0.749	0.663	0.640		0.900	0.775	0.792	
	σ_{a3}	0.264	0.227	0.219		0.365	0.285	0.280	
	σ_{b1}	1.120x10 ⁻²	6.818x10 ⁻²	3.844x10 ⁻²		1.139x10 ⁻²	7.676x10 ⁻³	3.910x10 ⁻³	
	σ_{b2}	3.675x10 ⁻²	3.305x10 ⁻²	3.165x10 ⁻²		4.490x10 ⁻²	4.325x10 ⁻²	4.144x10 ⁻²	
	σ_{b3}	3.178x10 ⁻²	3.011x10 ⁻²	2.892x10 ⁻²		3.395x10 ⁻²	3.319x10 ⁻²	3.551x10 ⁻²	

cases. The standard deviation values for the low and high noise/signal ratios were 0.1 and 5.0 for process I and 0.01 and 0.1 for processes II and III.

Computational expense is directly proportional to the amount of CPU time used and the two cases approximate the low and high ends of the cost of the four identification methods. Table 4.11 presents the results of the computational time estimation.

TABLE 4.11 Comparison of the Computational Time (seconds) Required for the Estimation Routines

estimation routine	process I	process II	process III	process I	process II	process III
	low noise/signal ratio, 100 data pairs			high noise/signal ratio, 500 data pairs		
STIME	5.558	5.509	5.613	6.419	6.291	6.382
ASTRM	5.688	2.269	7.013	5.818	4.894	2.955
IVOFF	0.434	0.446	1.529	1.246	0.971	3.528
UBIAS	0.332	0.393	0.756	0.841	0.669	1.848

4.4 SUMMARIZED RESULTS:

1) With all estimation routines and both noise/signal ratio types, increasing the length of the input/output sequences always increased the accuracy of the parameters estimated when processes I and II were tested.

2) For processes I and II with all estimation routines, increasing the noise/signal ratio always decreased the accuracy of the parameters estimated for the same number of data pairs.

3) Identification using generalized least squares did not provide converged process parameter values for all three processes with both noise/signal ratio types and all run lengths. Convergence was determined by CRITE, a user-set comparison variable in UNALIP which contained a value of 0.0001 for all tests.

4) For process III with all estimation routines, both noise/signal ratios and all run lengths, estimates of the denominator parameters are more reliable than estimates of the numerator parameters.

5) For process III with the generalized least squares estimation routine, increasing the number of data pairs does not necessarily mean an increase in the accuracy of the parameters estimated. This affected all the error measures. Similar behaviour was also noted by Isermann [2] and a suggestion that the autoregressive

process used to describe the residual sequence is not accurate enough was put forth. This accuracy problem stems from the truncation of an infinite autoregressive process.

6) For process III with the instrumental variables estimation routine and a low noise/signal ratio, the error measures were lowest at a run length of 300 data pairs. Similar behavior was also noted by Isermann [2] except that the standard deviation of the impulse response error does not follow a steady reduction in error magnitude. Isermann [2] suggested that this behaviour results from the model for process III.

7) For process III with the generalized least squares estimation routine and a low noise/signal ratio, the error measures were better than the other three estimation routines for a run length of 100 data pairs. As the number of data pairs or the noise/signal ratio increases, the comparison becomes less and less favourable. This could be due to the model for process III.

8) With all estimation routines, both noise/signal ratio types and all run lengths, the standard deviation of the mean squared impulse response error related to the mean squared true impulse function values for process III exhibits values similar to the other two processes even though the parameter estimates have large errors. Similar behavior was also noted by

Isermann [2]. A possible explanation is that process III has six unknown parameters while processes I and II have only four. The larger number of parameters would allow greater errors in the absolute values of the parameter while still estimating the system satisfactorily.

9) For process III with the generalized least squares and the unbiased estimator methods and run lengths of 300 and 500, all parameters estimated are almost identical, differing only in the fifth decimal place. No explanation for this can be given.

10) The instrumental variable and the unbiased estimator methods take the least computational time.

4.5 DISCUSSION

The generalized least squares method showed either slow or no convergence of the parameter estimates, thus indicating biased parameter estimates. This behaviour can be explained by remembering that the infinite autoregressive process to describe the residuals must be truncated to make it finite for computer computations. Thus the residuals are not described accurately enough producing a small bias. This appears not to have an effect for short run lengths.

When the noise/signal ratio increases, the instrumental variable method and the unbiased estimator method produce parameter estimates that converge very slowly. This could be the result of parameter approximation step size problems in the two routines. The parameter approximation step size determines how far the estimation of the parameter should move from the previous estimate of the parameter. Many of the difficulties encountered when the noise/signal ratio was high were due to the step sizes being too small even though the estimated parameters were not near the true parameters. This could be overcome by increasing the complexity of the parameter approximation step size reduction program code, but the increased complexity must be traded off with an increase in the computational time.

The maximum likelihood method appears to provide the most reliable parameter estimates for different numbers of data pairs, noise/signal ratio and process type but computational time must also be considered in rating the effectiveness of a particular estimation routine. Consideration of the computational time, the simulation results show that the preferred identification routines would be the unbiased estimator and instrumental variable methods, especially when the noise/signal ratio is low.

CHAPTER FIVE

PARAMETER ESTIMATION OF MODELS FOR PILOT PLANT UNITS

5.1 INTRODUCTION

This chapter contains the work done to estimate the coefficients of the pulse transfer functions of two pilot plant units. One unit was a liquid-liquid, concentric tube heat exchanger. The other was a binary distillation column with eight bubble cap trays.

Although both units exhibit nonlinear dynamic behaviour, the approximate behaviour of both pilot plant units can be represented by linear, time invariant, difference equations. Therefore the UNiversity of ALberta Identification Package (UNALIP) can be applied to identify pulse transfer functions that represent system behaviour.

5.2 DESCRIPTION OF THE PILOT PLANT UNITS

DISTILLATION COLUMN

The binary distillation column is 22.86 cm. in diameter with eight bubble cap trays and has glass walls for viewing. Tray spacing is 30.48 cm. All temperatures, flow rates, and compositions measured on the process are transmitted to an IBM 1800 digital computer and logged for data acquisition and/or control action by means of the standard Direct Digital Control (DDC) program [26]. The distillation column is equipped with analog controllers to control flow rates, reboiler and condenser levels and the feed and reflux temperatures. The flow controller set points are manipulated by means of the DDC program. A detailed description of the distillation column and the associated equipment is given by Svrcek [22] and Pacey [23]. Figure 5-1 is a simple schematic of the distillation column configuration.

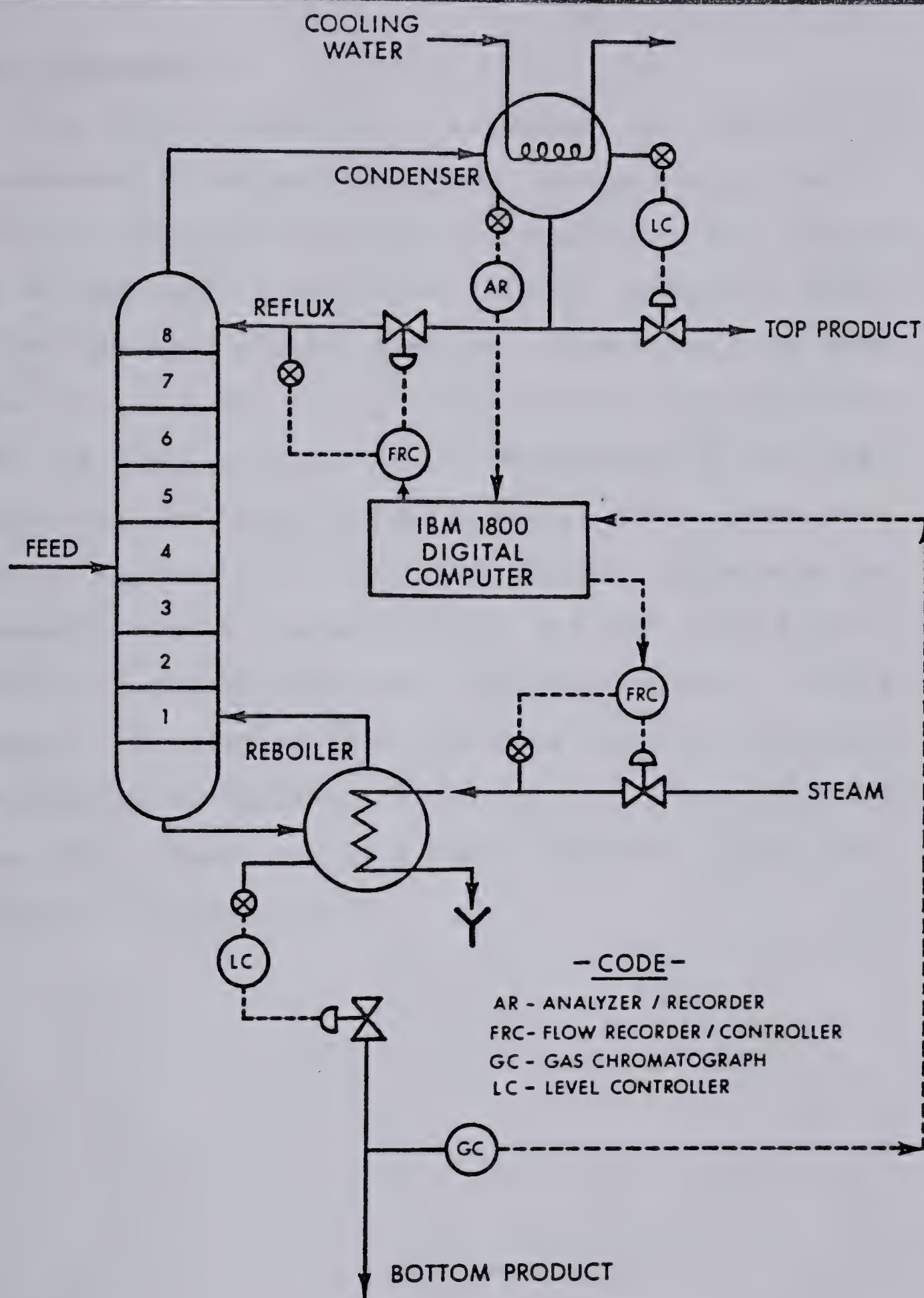


FIGURE 5-1
SIMPLE SCHEMATIC OF THE DISTILLATION COLUMN

HEAT EXCHANGER

The double pipe heat exchanger is operated in countercurrent manner with both streams being water. All temperatures and flow rates measured on the process are accumulated by the Direct Digital Control programs on the IBM 1800 digital computer, except that the tube side flow rate and the outlet temperatures for the tube side and shell side fluids are accumulated by the High Speed Data Acquisition (VCDAQ) package [24]. Tube side flow is regulated by a pneumatic control valve with the pneumatic signal manipulated by the IBM 1800 digital computer executing the heat exchanger dynamic testing program. A detailed description of the heat exchanger and associated equipment is given by Wildman [24] and Lees [25]. Figure 5-2 is a simple schematic of the heat exchanger configuration.

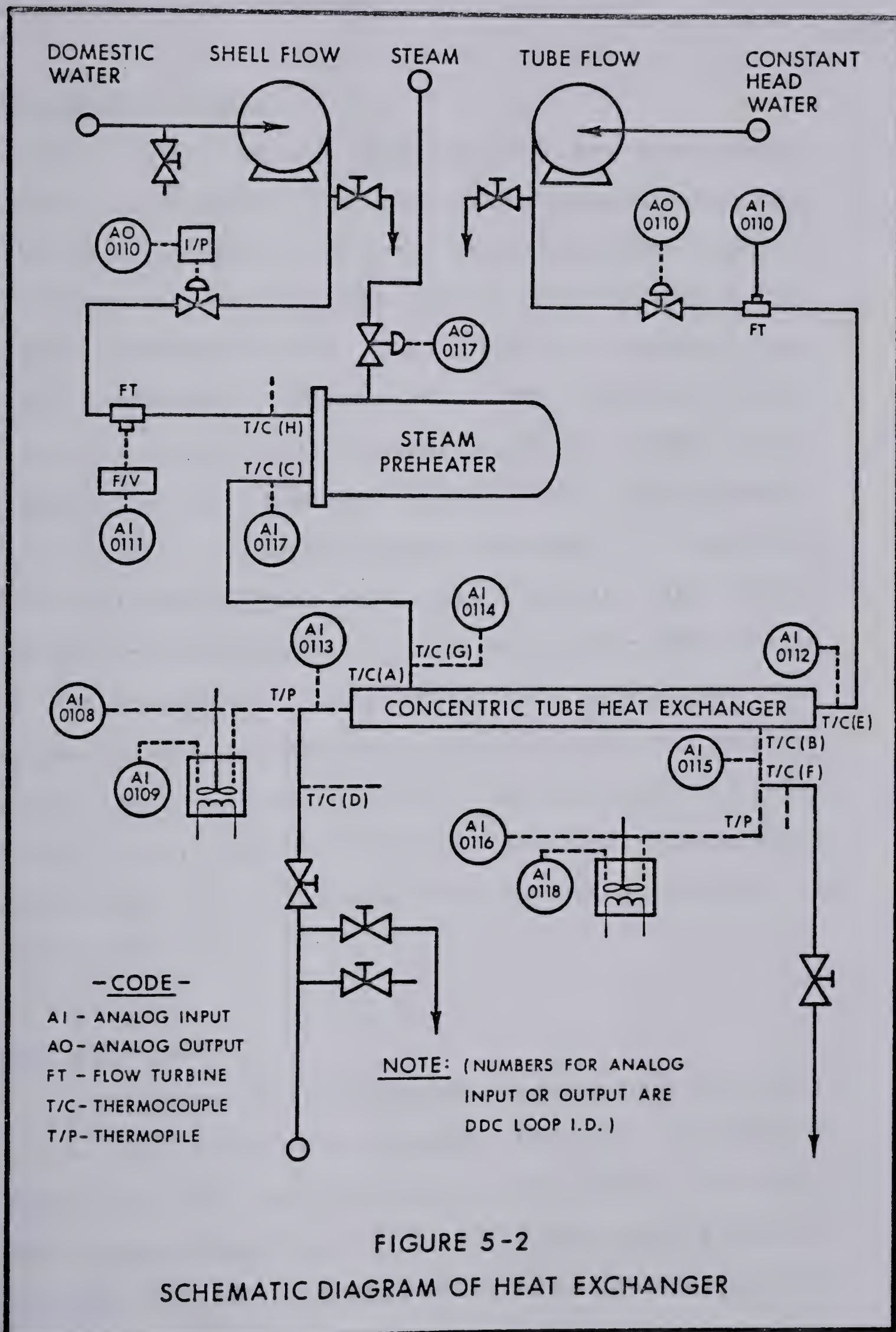


FIGURE 5-2

SCHEMATIC DIAGRAM OF HEAT EXCHANGER

5.3 EXPERIMENTAL PROCEDURE

DISTILLATION COLUMN

The object of the experiment on the distillation column was to obtain the four pulse transfer functions relating the inputs, the steam and reflux flow rates, to the outputs, the top and bottom compositions. The transfer functions would then be used in a control study being conducted by Bilec [26]. The steam and reflux flow rates were varied, using a pseudo random binary sequence as the forcing signal to the flow control valve. Pseudo random binary sequences of length 63, double 31 and 127 were used. The flow rates and the top and bottom compositions were logged via the DDC program of the IBM 1800. The resulting input/output data sequences were analyzed and the pulse transfer function coefficients determined by the identification routines ASTRM {3.10}, IVOFF {3.11} and UBIAS {3.12}. A detailed description of the experimental procedure is given by Bilec [26].

HEAT EXCHANGER

The object of the experiment on the heat exchanger was to obtain the pulse transfer function relating the input, the tube side flow rate, to the output, the shell side fluid outlet temperature. The tube side flow rate was varied, using a pseudo random binary sequence as the

forcing signal to the pneumatic control valve. The tube side flow rate and shell side fluid outlet temperature were logged by the VCDAQ program for subsequent analysis. The data was collected over 50 seconds at 10 points per second. Pseudo random binary sequences of lengths 255, with a clock interval of 1, and 31, with a clock interval of 10, were used. The clock interval is the number of sampling periods that the pseudo random binary sequence forcing signal to the pneumatic control valve maintains a certain state. The resulting input/output data sequences were analyzed to determine the pulse transfer functions using each of the identification techniques STIME {3.9}, ASTRM {3.10}, IVOFF {3.11} and UBIAS {3.12}. A detailed description of the experimental procedure is given in Appendix A.

5.4 RESULTS

DISTILLATION COLUMN

Tables 5.1 to 5.4 are tabulations of the system parameters for the identification runs involved. Appendix C has the plots of the input and output data for each test. The averaged pulse transfer functions obtained for the distillation column were

$$\begin{bmatrix} X_B(z) \\ X_D(z) \end{bmatrix} = \begin{bmatrix} \frac{-0.1675z^{-2}}{1 - 0.9185z^{-1}} & \frac{0.0630z^{-4}}{1 - 0.9456z^{-1}} \\ \frac{-0.0603z^{-1}}{1 - 0.8950z^{-1}} & \frac{0.0461z^{-1}}{1 - 0.7259z^{-1}} \end{bmatrix} \begin{bmatrix} ST(z) \\ RE(z) \end{bmatrix} \quad (5.1)$$

Using these averaged transfer functions, Bilec [26] developed s-domain approximations. The transfer functions he established were:

$$\begin{bmatrix} X_B(s) \\ X_D(s) \end{bmatrix} = \begin{bmatrix} \frac{-2.0552e^{-128s}}{1505.6s + 1} & \frac{1.1581e^{-384s}}{2288.3s + 1} \\ \frac{-0.5743}{1153.9s + 1} & \frac{0.1682}{399.6s + 1} \end{bmatrix} \begin{bmatrix} ST(s) \\ RE(s) \end{bmatrix} \quad (5.2)$$

TABLE 5.1: Comparison of Parameters Estimated by Three Identification Routines for the Steam and Bottom Composition Transfer Function

estimation routine	parameter	length of pseudo random binary sequence	disturbance
		63	DOUBLE 31 127
ASTRM	A	-0.9266	-0.9415
	B	-0.0885	-0.2784
IVOFF	A	-0.8010	-0.9404
	B	-0.1194	-0.2481
UBIAS	A	-0.9309	-0.9395
	B	-0.1200	-0.2482

TABLE 5.2: Comparison of Parameters Estimated by Three Identification Routines for the Steam and Top Composition Transfer Function

estimation routine	parameter	length of pseudo random binary sequence	disturbance
		63	127
		DOUBLE 31	
ASTRM	A	-0.9369	-0.9092
	B	-0.0574	-0.0569
IVOFF	A	-0.8578	-0.9075
	B	-0.0685	-0.0560
UBIAS	A	-0.8613	-0.9073
	B	-0.0685	-0.0560

TABLE 5.3: Comparison of Parameters Estimated by Three Identification Routines for the Reflux and Bottom Composition Transfer Function

estimation routine	parameter	length of pseudo random binary sequence	disturbance
		63	127
		DOUBLE 31	
ASTRM	A	-0.8953	-0.9515
	B	0.0383	0.0738
IVOFF	A	-0.8957	-0.9578
	B	0.0754	0.0756
UBIAS	A	-0.8994	-0.9594
	B	0.0754	0.0755

TABLE 5.4: Comparison of Parameters Estimated by Three Identification Routines for the Reflux and Top Composition Transfer Function

estimation routine	parameter	length of pseudo random binary sequence	disturbance
		63	127
ASTRM	A	-0.6039	-0.8026
	B	0.0082	0.0539
IVOFF	A	-0.7361	-0.7952
	B	0.0549	0.0426
UBIAS	A	-0.7365	-0.7952
	B	0.0549	0.0426

- where: 1) X_D is the top composition (weight percent methanol)
- 2) X_B is the bottom composition (weight percent methanol)
- 3) ST is the steam flow rate to the reboiler (g/s)
- 4) RE is the reflux flow rate (g/s)

Comparison of the gains of the estimated transfer functions with those established by step testing showed that there was large disagreement. Consequently in his control study, Bilec [26] used the gains from the open loop experimental tests and the time constants established by this identification study. The transfer functions that he used were:

$$\begin{bmatrix} X_B(s) \\ X_D(s) \end{bmatrix} = \begin{bmatrix} \frac{-8.8410e^{-128s}}{1037.4s + 1} & \frac{4.2180e^{-384s}}{1119.3s + 1} \\ \frac{-0.6310}{859.3s + 1} & \frac{0.5420}{418.1s + 1} \end{bmatrix} \begin{bmatrix} ST(s) \\ RE(s) \end{bmatrix} \quad (5.3)$$

After studying the problem of poor gain realization, it was noticed that the transfer functions with the poorest gains were the transfer functions that had the largest time constants. A simulated system was then studied to analyze the results from the different identification techniques to ascertain the cause of poor

estimation of the transfer function gains. The simulated system was

$$\frac{e^{-s}}{1 + 100s} \longrightarrow \frac{0.01z^{-1}}{1 - 0.99z^{-1}} \quad (5.4)$$

The actual gain of the simulated system is 1. The input of the simulated system was a pseudo random binary sequence of length 31. The clock interval used was 1. The total number of input points was 500. The output sequence was corrupted with white noise that had zero mean and a standard deviation of 0.1. The results of the experiment are given in Table 5.5.

TABLE 5.5 Identified Gains of Simulated System Using Each Estimation Technique

estimation routine	gain
STIME	0.185×10^{-2}
ASTRM	0.118×10^{-2}
IVOFF	0.155×10^{-2}
UBIAS	0.154×10^{-2}

The results in Table 5.5 confirmed the suspicion that there were no errors in the implementation of the identification techniques but that the difficulty was likely the particular choice of the pseudo random binary sequence input used for the tests. Consequently, it was decided to assess the effect of a change in the clock interval of the input pseudo random binary sequence upon the estimated gains. The clock interval was varied from 5 to 20 in predetermined stages to see if this would improve the gain estimates [2]. The length of the pseudo random binary sequence was maintained constant at a length of 31. The total number of data pairs was again selected at 500 and the calculated system output corrupted with zero mean and 0.1 standard deviation white noise. The results from identification performed on the resulting data pairs for particular clock intervals are given in Table 5.6.

Table 5.6 indicates that increasing the clock interval from 5 to 15 increases the accuracy of the estimated gains but further increases appear to have a negative influence.

TABLE 5.6 Effect of Clock Interval on the Estimated Gain of a Simulated System

clock interval	estimation routine	gain
5	STIME	0.146×10^{-1}
	ASTRM	1.038
	IVOFF	0.501
	UBIAS	0.516
10	STIME	0.494×10^{-1}
	ASTRM	1.038
	IVOFF	1.010
	UBIAS	0.957
15	STIME	1.100
	ASTRM	1.021
	IVOFF	1.137
	UBIAS	1.003
20	STIME	1.355
	ASTRM	0.954
	IVOFF	0.862
	UBIAS	0.862

HEAT EXCHANGER

The pulse transfer functions obtained when a pseudo random binary sequence of length 255 and clock interval of 1 sampling interval (0.1 sec.) was used for the input forcing function were:

1) Generalized Least Squares (STIME {3.9})

$$\frac{T_S(z)}{F_T(z)} = \frac{-0.001289z^{-11}}{1 - 0.9927z^{-1}} \quad (5.5)$$

where: 1) F_T is the tube side flow rate (liter/min)

2) T_S is the shell side fluid outlet temperature
(degrees C)

2) Maximum Likelihood (ASTRM {3.10})

$$\frac{T_S(z)}{F_T(z)} = \frac{-0.009052z^{-7}}{1 - 0.9963z^{-1}} \quad (5.6)$$

3) Instrumental Variables (IVOFF {3.11})

$$\frac{T_S(z)}{F_T(z)} = \frac{-0.01042z^{-12}}{1 - 0.9797z^{-1}} \quad (5.7)$$

4) Unbiased Estimator (UBIAS {3.12})

$$\frac{T_S(z)}{F_T(z)} = \frac{-0.01037z^{-12}}{1 - 0.9802z^{-1}} \quad (5.8)$$

The input forcing function values are generated between zero and plus or minus one. These values are then converted to controller output values acceptable to the control value. The initial steady state tube flow rate was 22.7 l/min. The input forcing function would cause the tube flow rate to vary ± 11.35 l/min. depending on the sign of the input forcing function.

The pulse transfer functions obtained when a pseudo random binary sequence of length 31 and clock interval of 10 sampling intervals (1.0 sec.) was used for the input forcing function were:

1) Generalized Least Squares (STIME {3.9})

$$\frac{T_S(z)}{F_T(z)} = \frac{-0.009326z^{-1}}{1 - 0.9964z^{-1}} \quad (5.9)$$

2) Maximum Likelihood (ASTRM {3.10})

$$\frac{T_S(z)}{F_T(z)} = \frac{-0.01855z^{-16}}{1 - 0.9843z^{-1}} \quad (5.10)$$

3) Instrumental Variables (IVOFF {3.11})

$$\frac{T_S(z)}{F_T(z)} = \frac{-0.01855z^{-16}}{1 - 0.9843z^{-1}} \quad (5.11)$$

4) Unbiased Estimators (UBIAS)

$$\frac{T_S(z)}{F_T(z)} = \frac{-0.01846z^{-16}}{1 - 0.9846z^{-1}} \quad (5.12)$$

The input forcing function values are generated between zero and plus or minus one. These values are then converted to controller output values acceptable to the control valve. The initial steady state tube flow rate

was 22.7 l/min. The input forcing function would cause the tube flow rate to vary ± 11.35 l/min. depending on the sign of the input forcing function.

Appendix B contains the plots of the experimental input and output data for each test run. The steady state energy balance, initial test conditions and a summary of the psuedo random binary sequence conditions are given in Appendix A.

Wildman [24] determined continuous transfer functions by pulse testing using four different input pulse shapes. Averaging of the time constants (time delays) and gain values gives the transfer function

$$\frac{T_S(s)}{F_T(s)} = \frac{-1.07e^{-1.14s}}{1 + 3.15s} \quad (5.13)$$

For the sampling time of 0.05 seconds, the pulse transfer function is

$$\frac{T_S(z)}{F_T(z)} = \frac{-0.01674z^{-11}}{1 - 0.9843z^{-1}} \quad (5.14)$$

As can be seen, the agreement between the average transfer function representation established by Wildman and the transfer functions estimated by the ASTRM {3.10}, IVOFF {3.11} and UBIAS {3.12} routines are good

for both experimental runs. However the best estimates resulted using the psuedo random binary sequence of length 31 and clock interval of 10 sampling intervals (1.0 sec.). Excellent gain estimates were obtained.

It is to be noted that the generalized least squares estimated model was not consistent with the models established using the other three identification routines nor with the model Wildman determined [24]. Similar behavior was noted in Chapter Four.

These results are thus consistent with the model estimates in Chapter 4 established using the different routines.

5.5 DISCUSSION

The estimation of the pulse transfer functions by the UNiversity of ALberta Identification Package (UNALIP), in general, was acceptable. The generalized least squares identification (STIME {3.9}) routine was consistently poorer than the other three routines, producing a time delay and gain values that were unacceptable. This is the reason that only the maximum likelihood (ASTRM {3.10}), instrumental variables (IVOFF {3.11}) and unbiased estimator (UBIAS {3.12}) routines were used to estimate pulse transfer functions for the distillation column.

For specific transfer functions, the estimation routines for individual disturbance types produced results that were very close to each other. For different disturbance types, the results were not always consistent. Pseudo random binary sequence input signal lengths of 63 and double 31 produced comparable results but the estimates from the pseudo random binary sequence input signal of length of 127 were not comparable to the other tests. On the basis of the simulation results in Chapter Four it was concluded that the results from the pseudo random binary sequence length of 127 should be used because generally parameter estimate accuracy increased if the number of data pairs increased.

A possible reason for the poor gain estimation for the distillation column is that for systems with large time constants, the clock interval must be large so that the process is able to 'see' the input disturbance. A criterion for a good input sequence is that it excites all possible states of the process [27]. This was probably not the case for the distillation column tests. Unfortunately runs to test this were not done on the distillation column but a type of check was accomplished on the heat exchanger. A pseudo random binary sequence of length 31 and clock interval of 10 gave better results than the pseudo random binary sequence of length 255 and clock interval of 1. Also, the estimates of the heat exchanger transfer function were very similar for the estimation routines ASTRM {3.10}, IVOFF {3.11} and UBIAS {3.12} which is consistent with the previous simulation results.

There was good agreement between the estimated time delays for the distillation column pulse transfer functions and the time delays determined by Bilec [26] from step testing. This was not the case for the heat exchanger but from the range in the values of the time constants determined by Wildman [24], it is reasonable to conclude that the estimation of small time delays is difficult for a real process. Wildman [24] estimated the time delay to be equal to 11 sample intervals or

0.55 seconds, which is close to the 16 sample intervals or 1.60 second time delay estimated in this study.

It can be concluded that when estimating the pulse transfer function of an actual process that:

1) as many tests as possible with different input sequences should be performed using

- a) different clock intervals
- b) different length of input sequences
- c) different types of input sequences
- d) different magnitude of input sequence changes from steady state

2) as many identification methods as possible should be applied to the data

3) in deciding upon the final estimated pulse transfer function from the functions provided by the identification routines, care should be taken to compare the estimates from various routines and use only those estimates that experience dictates are reasonable for the process under consideration to obtain an average transfer function.

Appendix D is a collection of guidelines for performing identification of an unknown system. These guidelines were established during the work on the pilot plant units.

CHAPTER SIX

CONCLUSIONS AND RECOMMENDATIONS

The following conclusions can be drawn from this study:

- 1) On the basis of analysis of the experimental tests and generated data sequences, the preferred estimation routines were the instrumental variable and the unbiased estimator methods, considering both parameter estimation accuracy and computer cost.
- 2) The maximum likelihood technique, provided the most accurate parameter estimation for the tests using generated input/output system values.
- 3) The instrumental variable and the unbiased estimator methods provided the best parameter estimates using the experimental data from both pilot plant units.
- 4) Increasing the number of input/output data pairs used by the estimation routines does not always increase the accuracy of the parameter estimates. The order of the system being studied has an effect on the accuracy of the parameter estimates.

- 5) Increasing the noise/signal ratio for the three simulated processes decreased the accuracy of the parameter estimate for all estimation routines.
- 6) The accuracy of the parameter estimates decreases for all estimation routines as the order of the generated system increases.
- 7) In a first order system, increasing the clock interval of the input sequence increases the accuracy of the estimated gain if the time constant is large compared to the sampling interval. The clock interval must be varied until gain estimates are comparable to open loop values of the gain. It is possible to increase the clock interval of the input signal too much thus giving poor estimates of the gains.

The following are some recommendations for future work:

- 1) The UNiversity of ALberta Identification Package is now implemented on the University of Albert's Amdahl 470V/6 digital computer. The package should be implemented on the Department of Chemical Engineering's HP1000 distributed network digital computer. This would allow easy access to the

package for engineering studies conducted in the department. One disadvantage would be the reduced response time obtained by the package on the HP1000.

- 2) All the estimation routines in UNALIP are of the off-line type, an improvement in the arrangement would be to convert the routines to an on-line type for implementation on the HP1000 to allow real time model estimations to be included in control studies.
- 3) All the estimation routines can be converted to recursive forms which would save computational time and save core memory space. This would make conversion of UNALIP from the Amdahl mainframe to the HP minicomputer more feasible.
- 4) Some theoretical work in generalized least squares could be done on the problem of convergence of the parameter estimates to the neighbourhood of their true values [8]. The generalized least squares routine could then either be changed to promote convergence or eliminated as an estimation routine in UNALIP since it is currently doing an unsatisfactory job.
- 5) More testing of the identification package should be carried out, especially the use of high pass filters

and differencing used in the routine DMOVE. The testing would determine if these options help in the estimation process and if so, how do they aid improved estimation.

- 6) When the identification package is implemented on the HP1000 digital distributed digital computer network, it is recommended that a time series analysis program be included since there is no program of this type currently on the system.
- 7) An investigation into the effect of the clock interval on the estimation process should be carried out. This study could look into:
 - 1) what value of the clock interval will give the best estimates for different processes
 - 2) does the clock interval affect the estimation of the process's time delay
 - 3) when should the clock interval be altered

BIBLIOGRAPHY

- 1) KOGEKAR, H. V.: 'An Application Of Time Series Analysis for the Identification and Control of a Double Effect Evaporator', M. Sc. Thesis, Department of Chemical Engineering, University of Alberta (1977).
- 2) Isermann, R., Baur, U., Bamberger, W., Kneppo, P., and Siebert, H.: 'Comparison of Six On-line Identification and Parameter Estimation Methods', Automatica, 10, 81-103 (1974).
- 3) Astrom, K. J., Bohlin, T. and Wensmark, S.: 'Automatic Construction of Linear Stochastic Dynamic Models for Stationary Industrial Processes with Random Disturbances using Operating Records', IBM Nordic Laboratory (Report TP18.150), Solna, Sweden (1965).
- 4) Astrom, K. J. and Bohlin, T.: 'Numerical Identification of Linear Dynamic Systems from Normal Operating Records', Proc. 2nd IFAC Symposium on Theory of Self-Adaptive Systems, Teddington, England, 76-111, (1965).

5) Sage, A. P. and Melsa, J. L.: Estimation Theory with Application to Communications and Control, McGraw-Hill, New York (1971).

6) Strejc, V.: 'On Least Squares and Regression Methods in Identification: A Survey of Progresses and Trends', Seventh IFAC Triennial World Congress, paper 45A.2, Helsinki, Finland (1978).

7) Clarke, D. W.: 'Generalized Least Squares Estimation of the Parameters of a Dynamic Model', First IFAC Symposium on Identification, paper 3-17, Prague, Czechoslovakia, (1967).

8) Hastings-James, R. and Sage, M. W.: 'Recursive Generalized Least Squares Procedure for On-line Identification of Process Parameters', Proc. IEEE, 116, no. 12, 2057-2062 (1969).

9) Eykhoff, P.: System Identification--Parameter and State Estimation, Wiley, New York (1974).

- 10) Wozencraft, J. M. and Jacobs, T. M.:
Principles of Communication Engineering,
Wiley, New York (1965).
- 11) Himmelblau, D. M.: Process Analysis by
Statistical Methods, Wiley, New York (1970).
- 12) Levenberg, K.: 'A Method for the Solution
of Certain Nonlinear Problems in Least
Squares', Quart. Appl. Math., 2, 164-168
(1944).
- 13) Young, P. C.: 'An Instrumental Variable
Method for Real-Time Identification of a Noisy
Process', Automatica, 6, 271-287 (1970).
- 14) Smets, A. J.: 'The Instrumental Variable
Method and Related Identification Schemes',
TH Report 70-E-15, Dept. of Electrical
Engineering, Eindhoven U. of Tech. (1975).
- 15) Wong, K. Y. and Polak, E.: 'Identification
of Linear Discrete Time Systems using the
Instrumental Variable Method', IEEE Trans. on
Auto. Control, AC-12, no. 6, 707-718 (1967).

16) James, P. N., Souter, P. and Dixon, D. C.:
'Suboptimal Estimates of the Parameters of
Discrete Systems in the Presence of Correlated
Noise', Electronics Letters, 8, no. 16,
411-412 (1972).

17) Melsa, J. L. and Sage, A. P.: An
Introduction to Probability and Stochastic
Processes, Prentice Hall, New Jersey (1973).

18) 'APL Forecasting using Times Series
Analysis', Program Number: 5796-PFX, IBM
Corp., Systems Products Div., Dept. 758, Bldg.
501-048, East Fishkill, New York (1975).

19) System/360 Scientific Subroutine Package,
(360A-CW-03X), Version III, IBM Corp.,
Technical Publications Dept., 112 E. Post
Road, White Plains, New York (sixth edition).

20) 'The Michigan Terminal System, Volume 4:
Applications Subroutines', University of
Alberta, Edmonton, Alberta (1972).

- 21) Berka, V. F. and Jackson, G. R.: 'MTS FORTRAN User's Guide', Computing Services, University of Alberta, Edmonton, Alberta (1974).
- 22) Svrcek, W. Y.: 'Binary Distillation Column Dynamics', Ph.D. Thesis, Department of Chemical Engineering, University of Alberta (1967).
- 23) Pacey, W. C.: 'Control of a Binary Distillation Column: An Experimental Evaluation of Feedforward and Combined Feedforward-Feedback Control Schemes', M.Sc. Thesis, Department of Chemical Engineering, University of ALberta (1973).
- 24) Wildman, T. A.: 'On-Line Pulse Testing', M.Sc. Thesis, Department of Chemical Engineering, University of Alberta (1969).
- 25) Lees, R. S.: 'Pulse Testing with an On-Line Digital Computer', M.Sc. Thesis, Department of Chemical Engineering, University of Alberta (1969).

26) Bilec, R.: Unpublished M.Sc. Thesis,
Department of Chemical Engineering, University
of Alberta.

27) Godfrey, K. R.: 'The Theory of the
Correlation Method of Dynamic Analysis and its
Application to Industrial Processes and
Nuclear Power Plant', Measurement and Control,
2, T65-T72 (1969).

28) Saridis, G. N.: 'Comparison of Six On-Line
Identification Algorithms', Automatica, 10,
69-74 (1974).

29) Kubacek, L. and Kubackova, L.: 'The
Present Approach to the Study of the
Least-Squares Method', Studia Geoph. et Geod.,
22, 140-147 (1978).

APPENDIX A

This appendix contains the detailed description of the experimental procedure to obtain the data from the heat exchanger. This data was used by the estimation routines in UNALIP to provide the results discussed in Chapter Five.

Experimental Data for Test A

Forcing Function - pseudo random binary sequence
 - 255 cycles/pulse width
 - clock interval equal to one
 sampling interval

Response Function - shell liquid outlet temperature

Flow Conditions - countercurrent, water-water

Heat Balance Data

Tube fluid flow (l/min)	- 22.7
Shell fluid flow (l/min)	- 45.4
Tube Fluid inlet temperature (degrees C)	- 12.3
Tube fluid outlet temperature (degrees C)	- 28.9
Shell fluid inlet temperature (degrees C)	- 77.4
Shell fluid outlet temperature (degrees C)	- 68.7

Experimental Data

Duration of forcing function (seconds)	- 50
Total time for data accumulation (seconds)	- 50
Number of data points accumulated per second	- 10
Amplitude of the forcing function (percent)	- 5

Experimental Data for Test B

Forcing Function - pseudo random binary sequence
 - 31 cycles/pulse width
 - clock interval equal to ten
 sampling intervals

Response Function - shell liquid outlet temperature

Flow Conditions - countercurrent, water-water

Heat Balance Data

Tube fluid flow (l/min)	- 22.7
Shell fluid flow (l/min)	- 45.4
Tube Fluid inlet temperature (degrees C)	- 12.3
Tube fluid outlet temperature (degrees C)	- 27.4
Shell fluid inlet temperature (degrees C)	- 76.7
Shell fluid outlet temperature (degrees C)	- 69.0

Experimental Data

Duration of forcing function (seconds)	- 50
Total time for data accumulation (seconds)	- 50
Number of data points accumulated per second	- 10
Amplitude of the forcing function (percent)	- 5

APPENDIX B

Appendix B contains the plots of the input and output data for each test conducted on the heat exchanger.

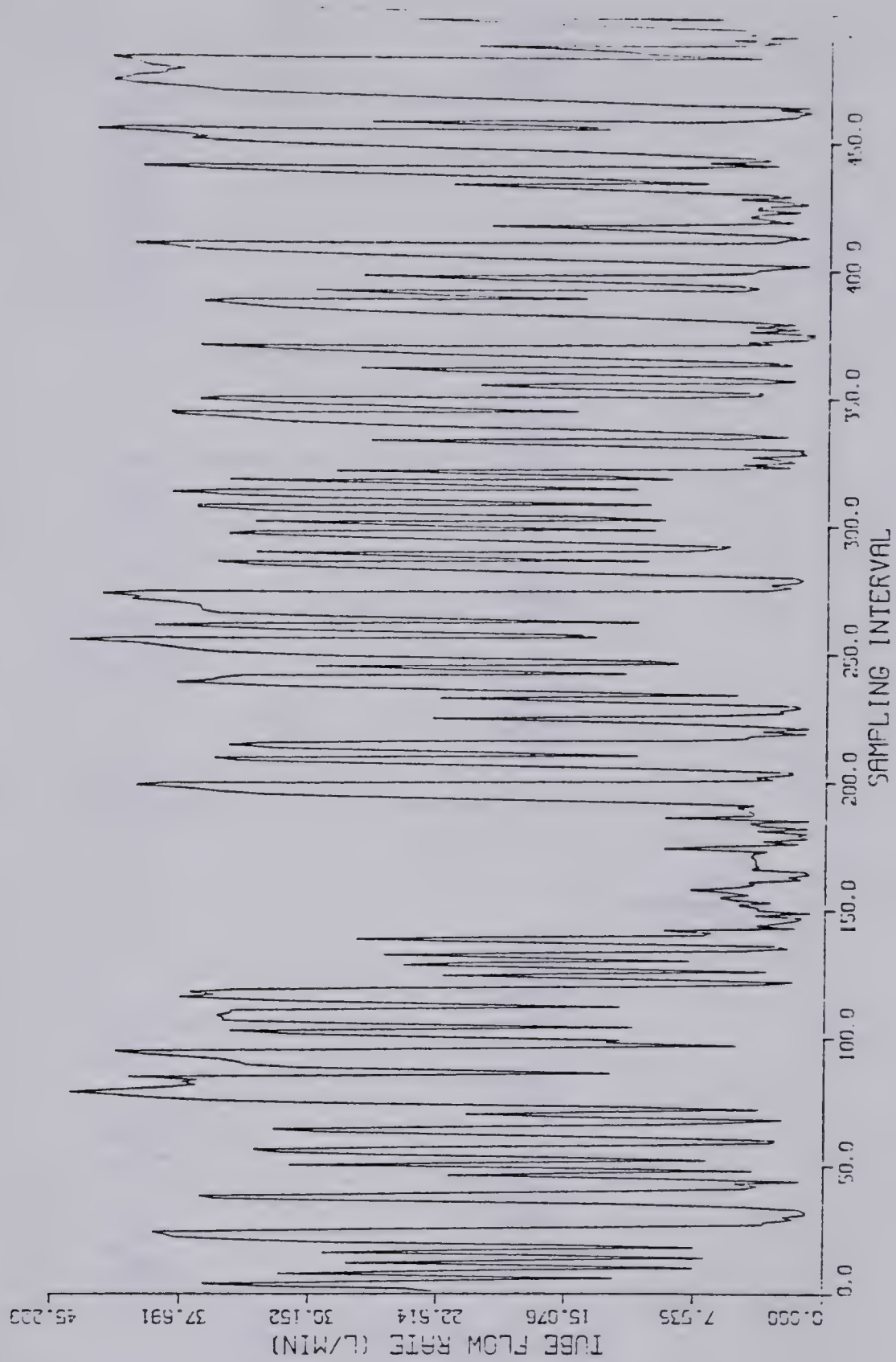


Figure B-1 Tube Flow Rate Response for Test A

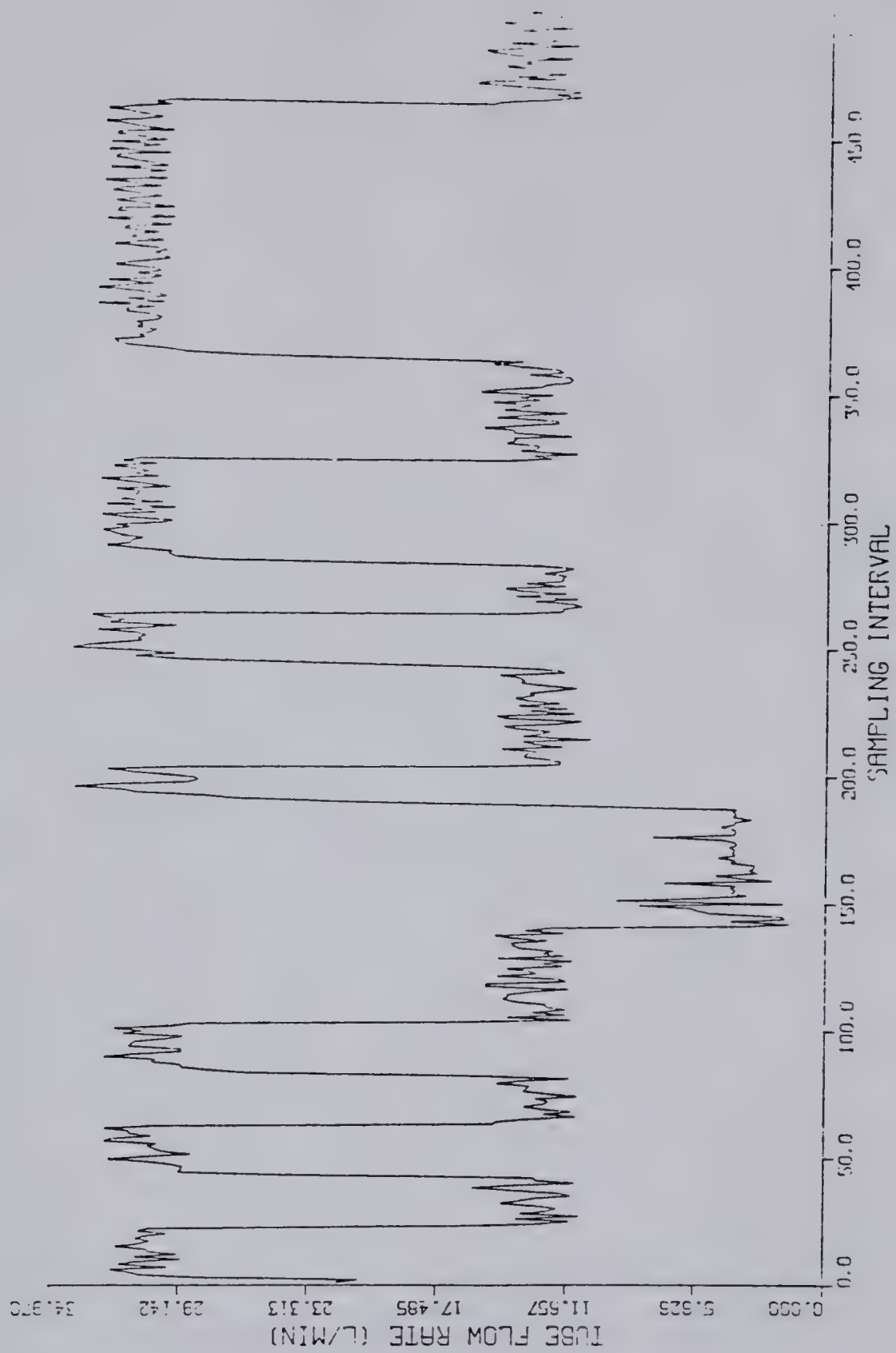


Figure B-2 Tube Flow Rate Response for Test B

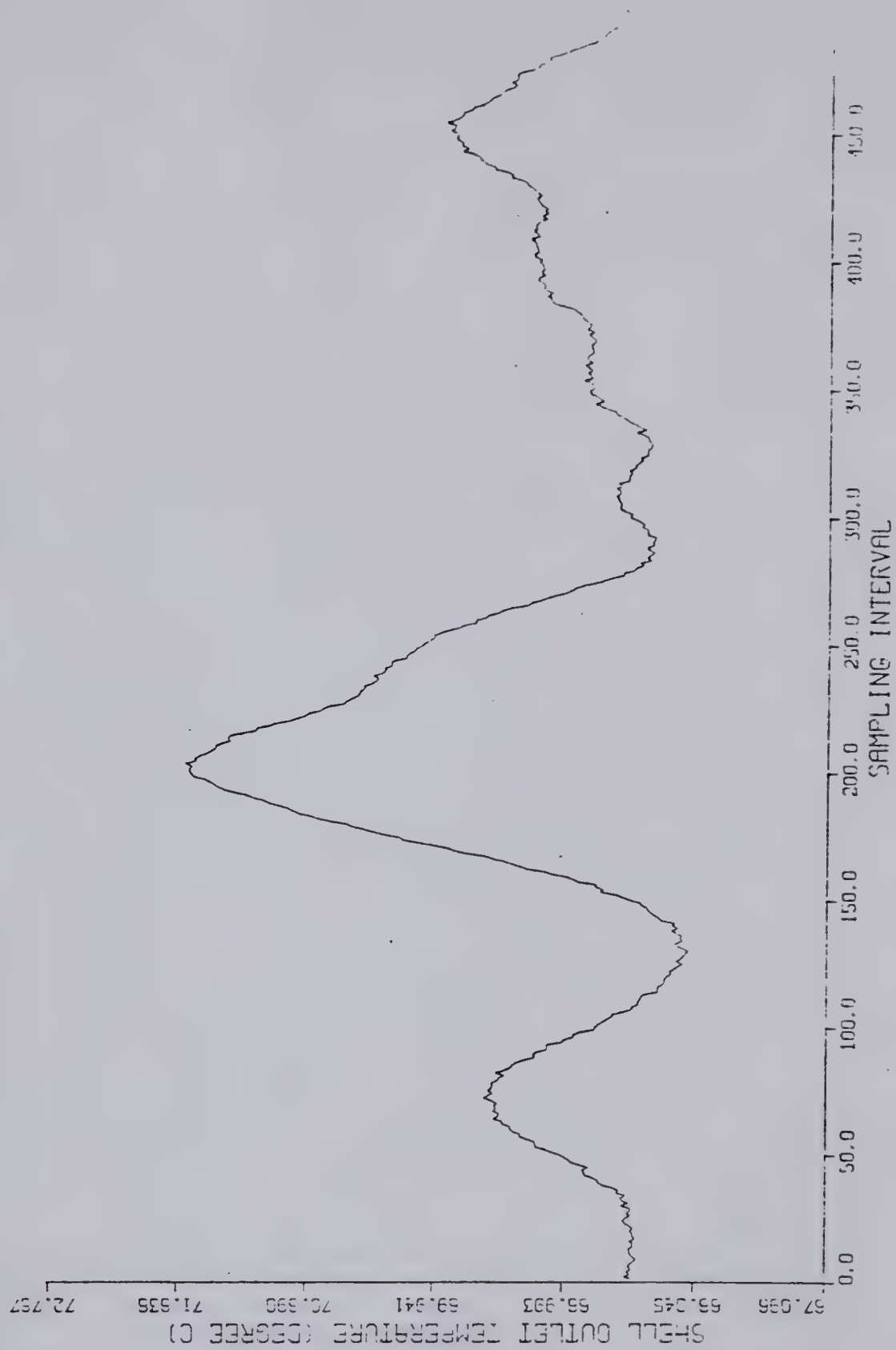


Figure B-3 Shell Outlet Temperature Response for Test A

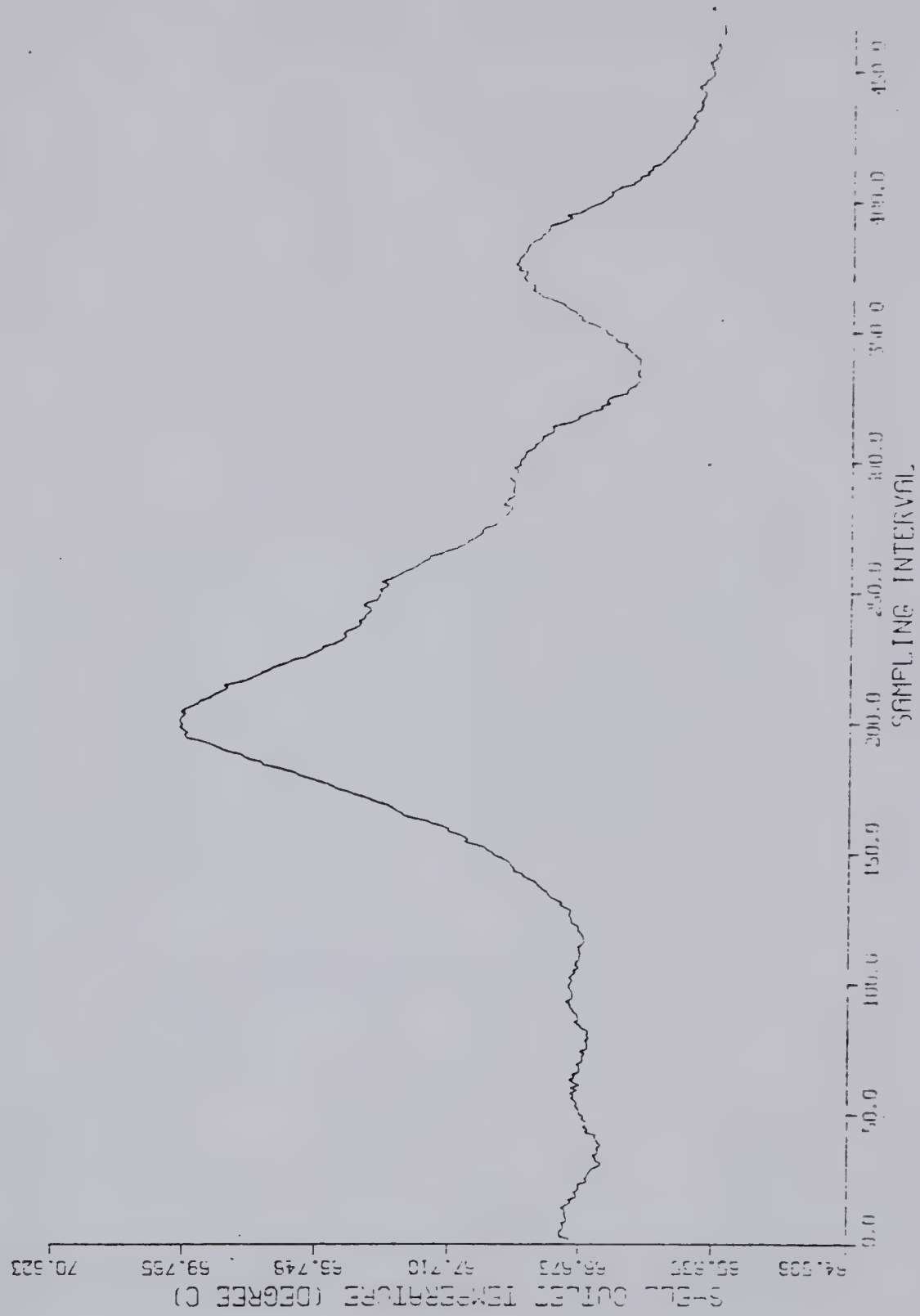


Figure B-4 Shell Outlet Temperature Response for Test B

APPENDIX C

Appendix C contains the plots of the input and output data for each test conducted on the distillation column.

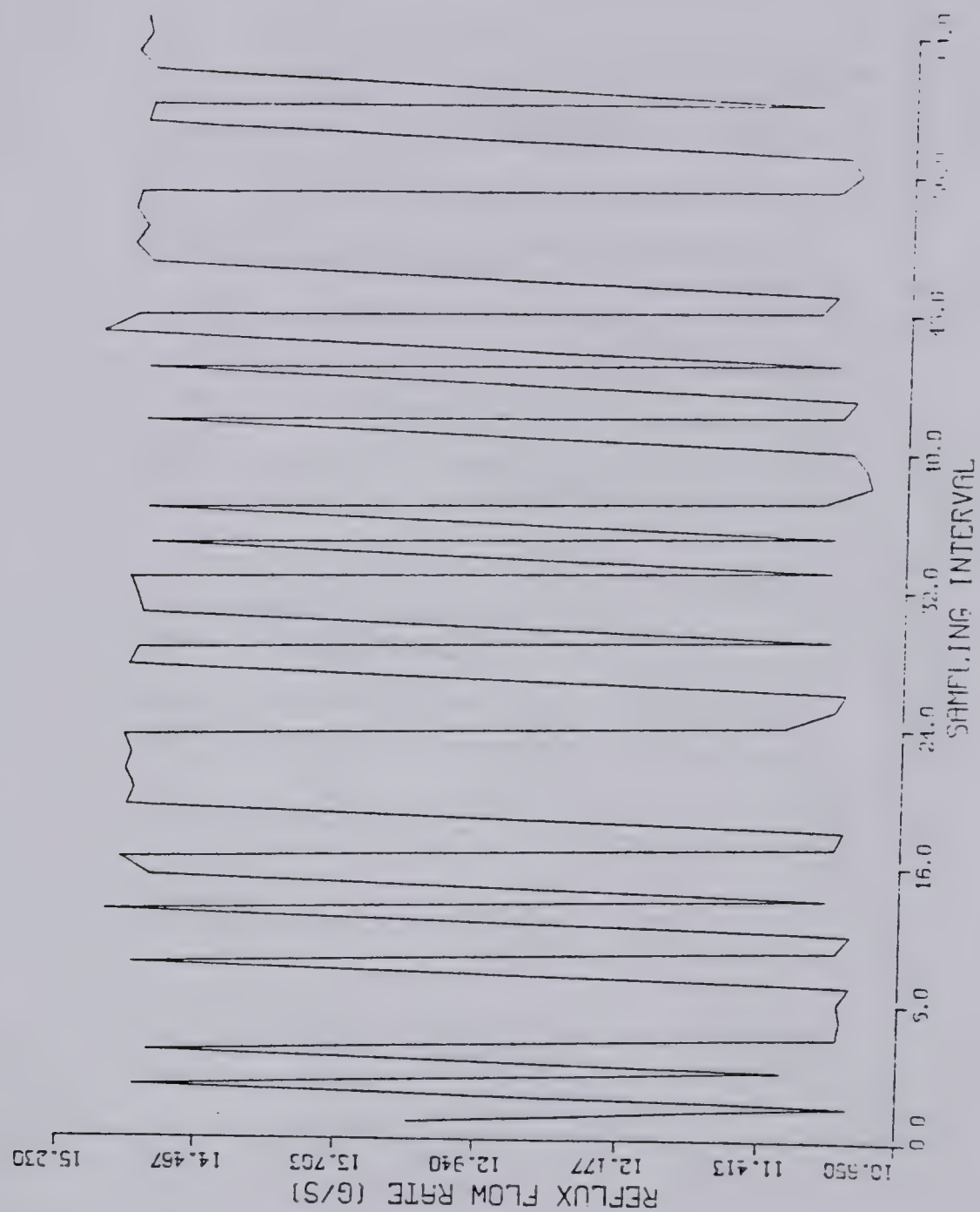


Figure C-1 Reflux Flow Rate Response for PRBS Disturbance of Length 63

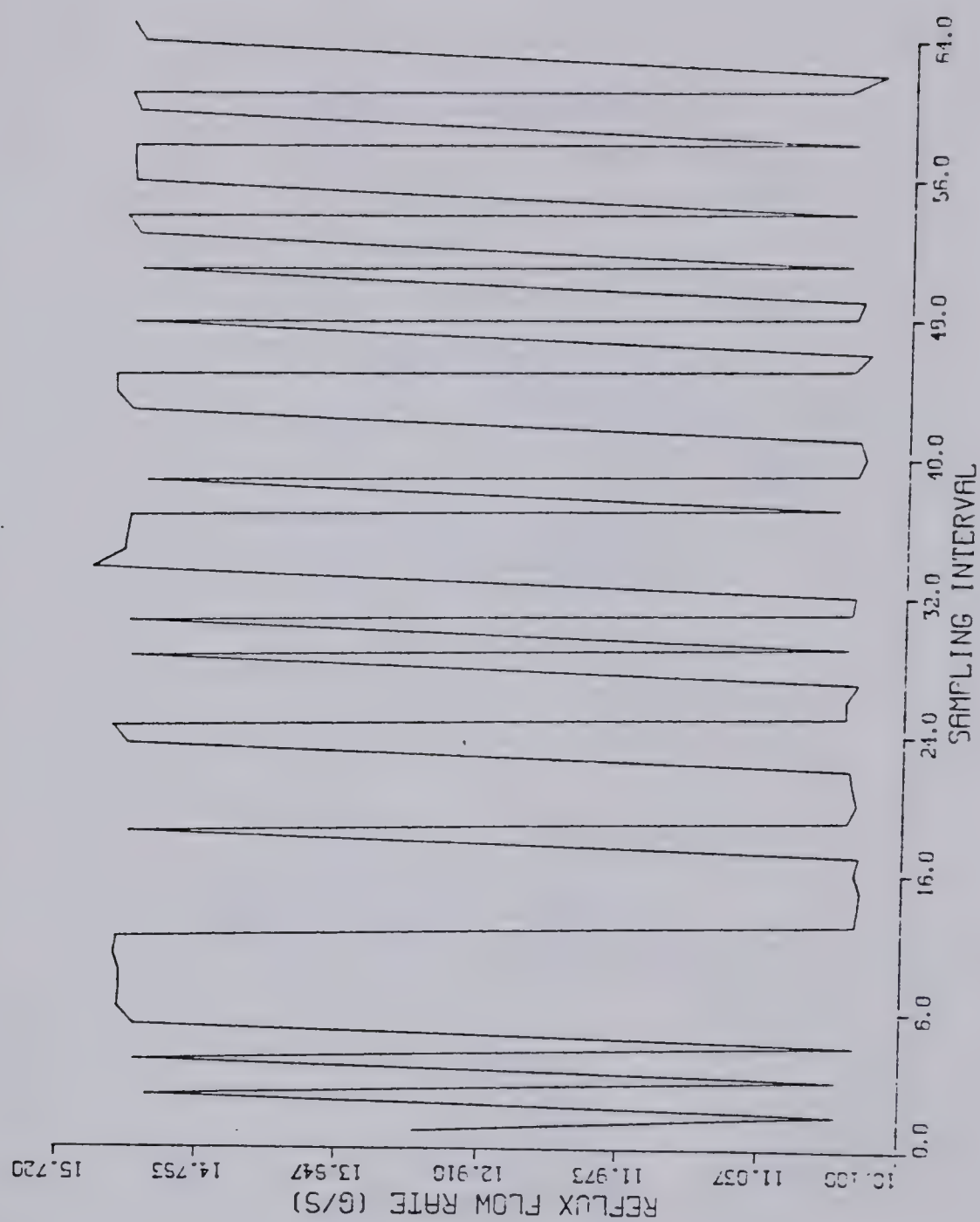


Figure C-2 Reflux Flow Rate Response for PRBS
Disturbance of Length Double 31

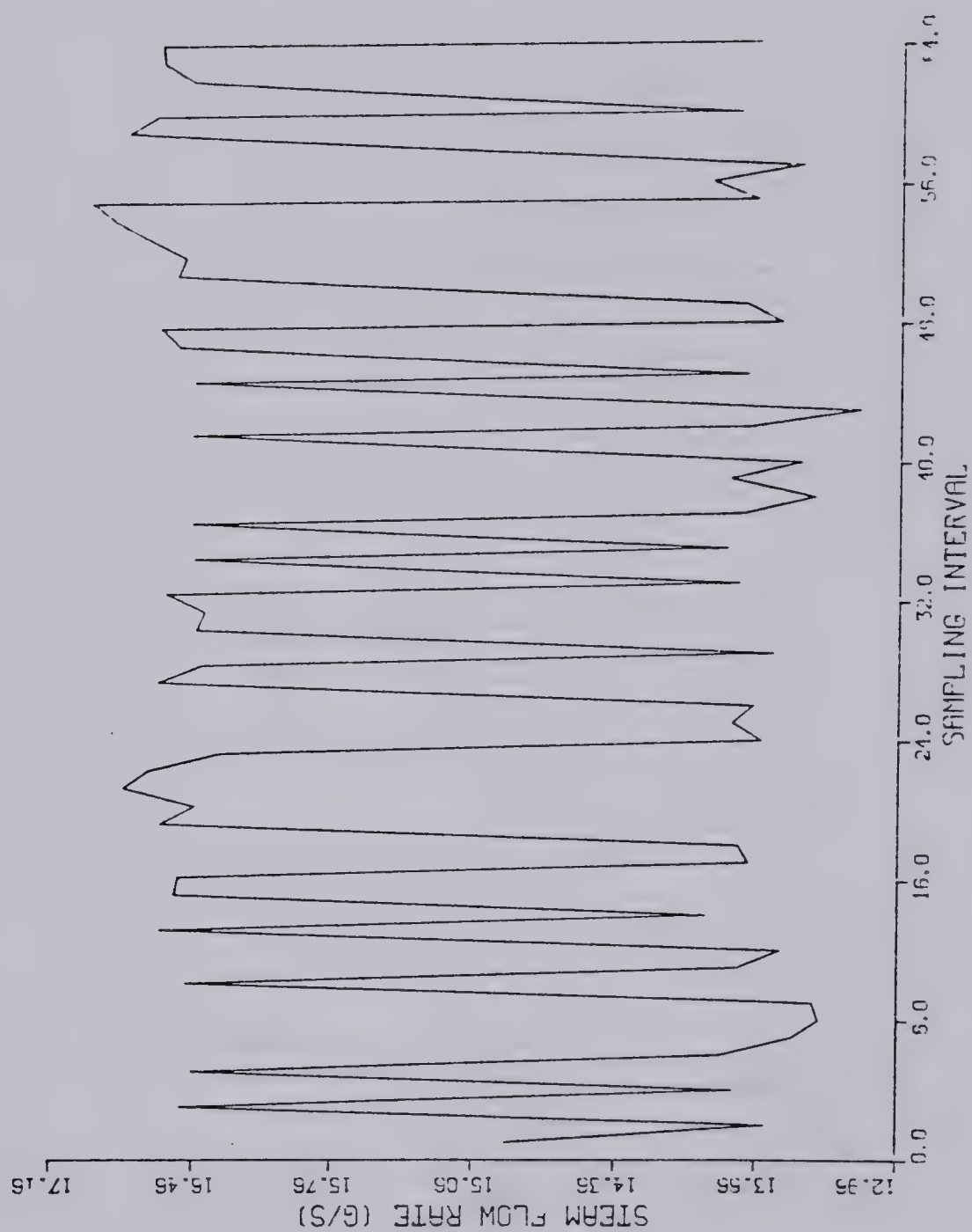


Figure C-3 Steam Flow Rate Response for PRBS
Disturbance of Length 63

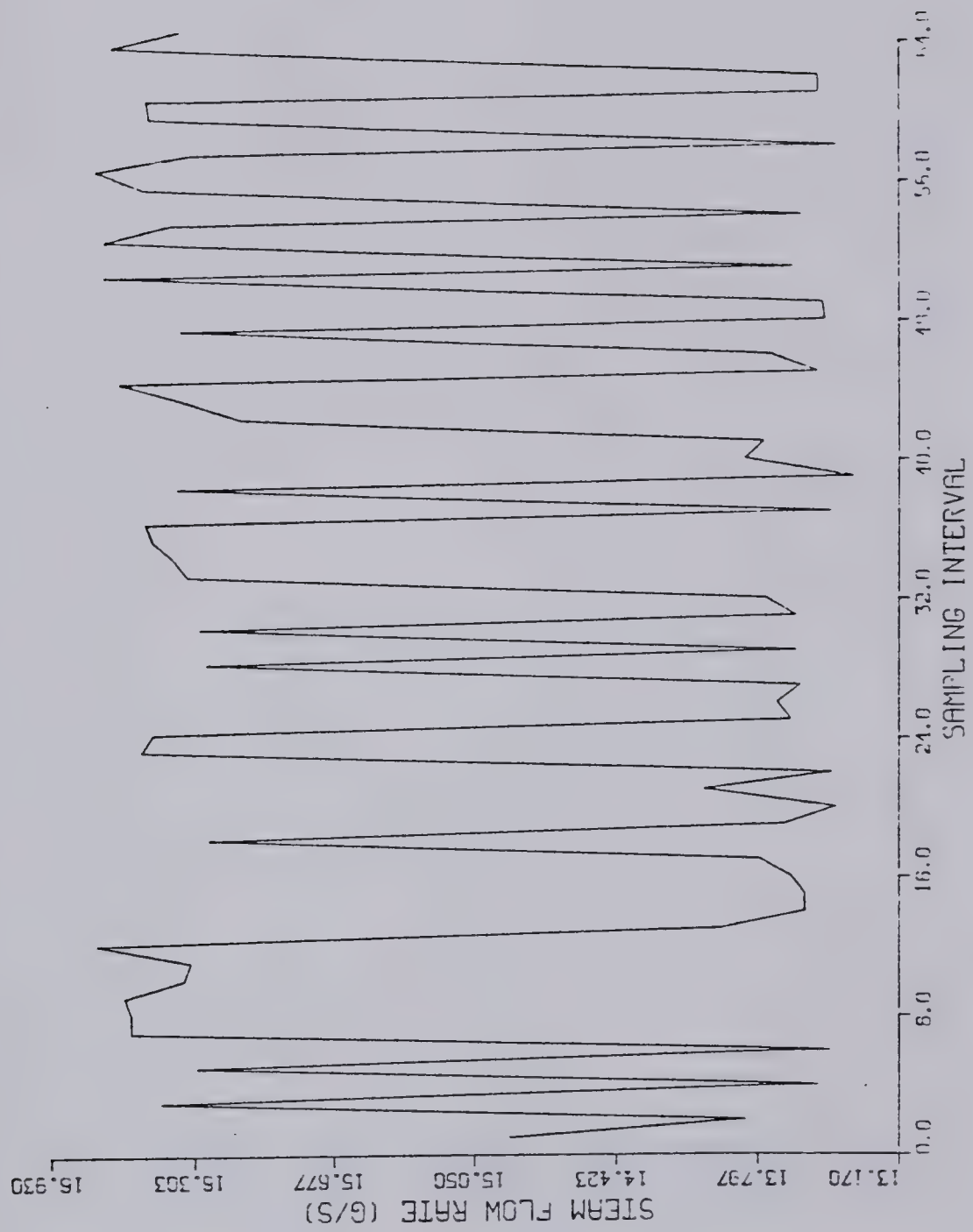


Figure C-4 Steam Flow Rate Response for PRBS
Disturbance of Length Double 31

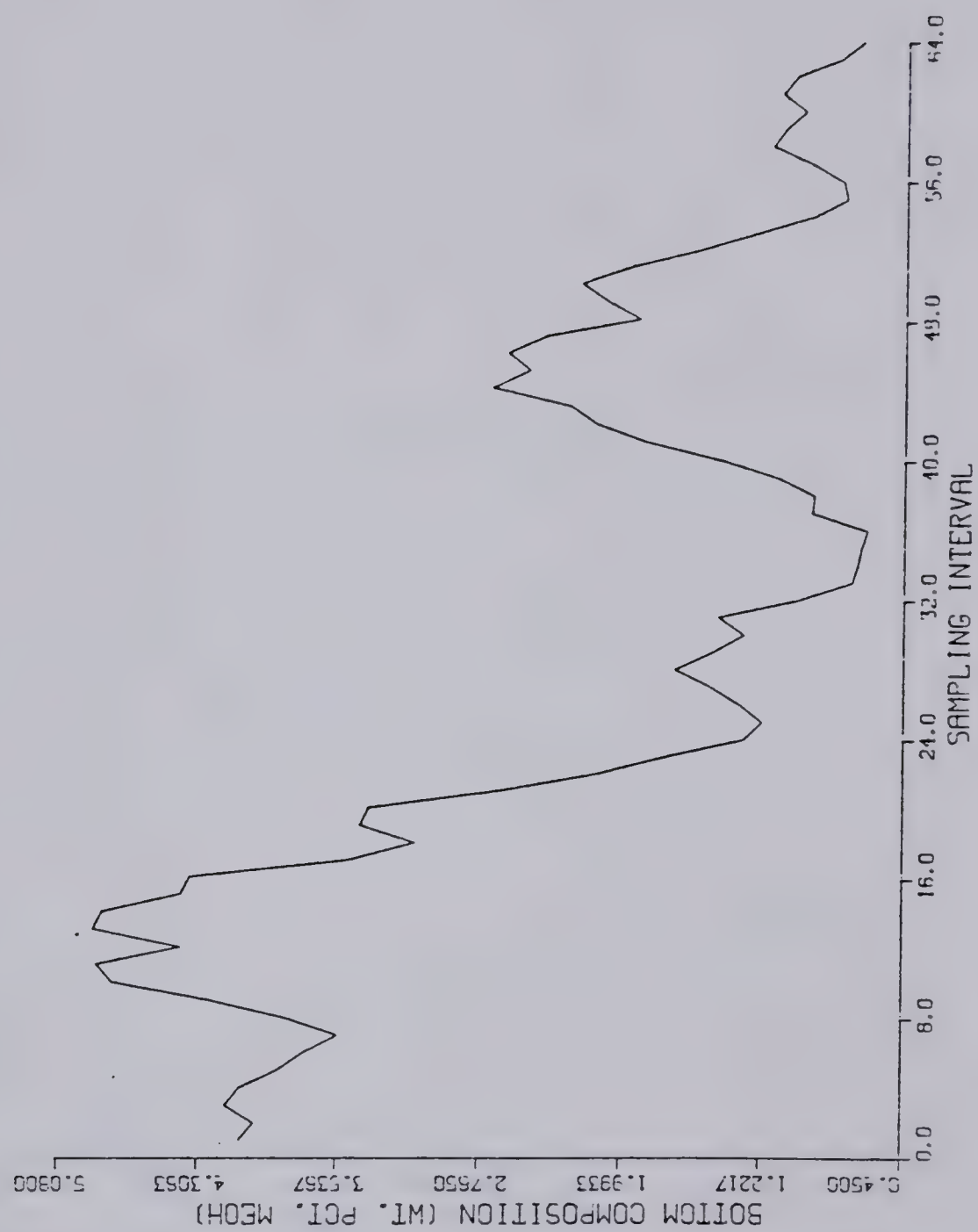


Figure C-5 Bottom Composition Response for PRBS Disturbance of Length 63 using Steam

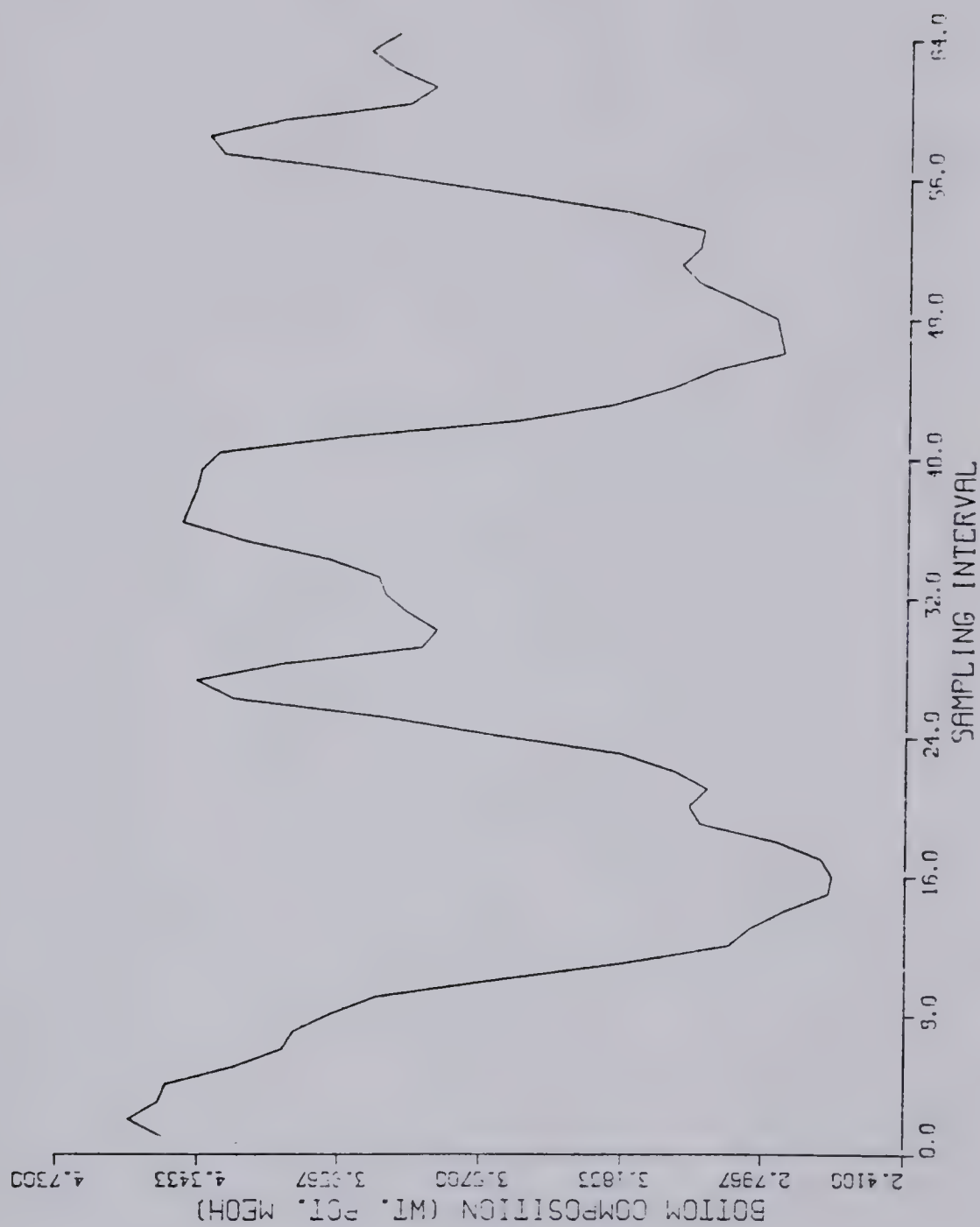


Figure C-6 Bottom Composition Response for PRBS Disturbance of Length 63 using Reflux

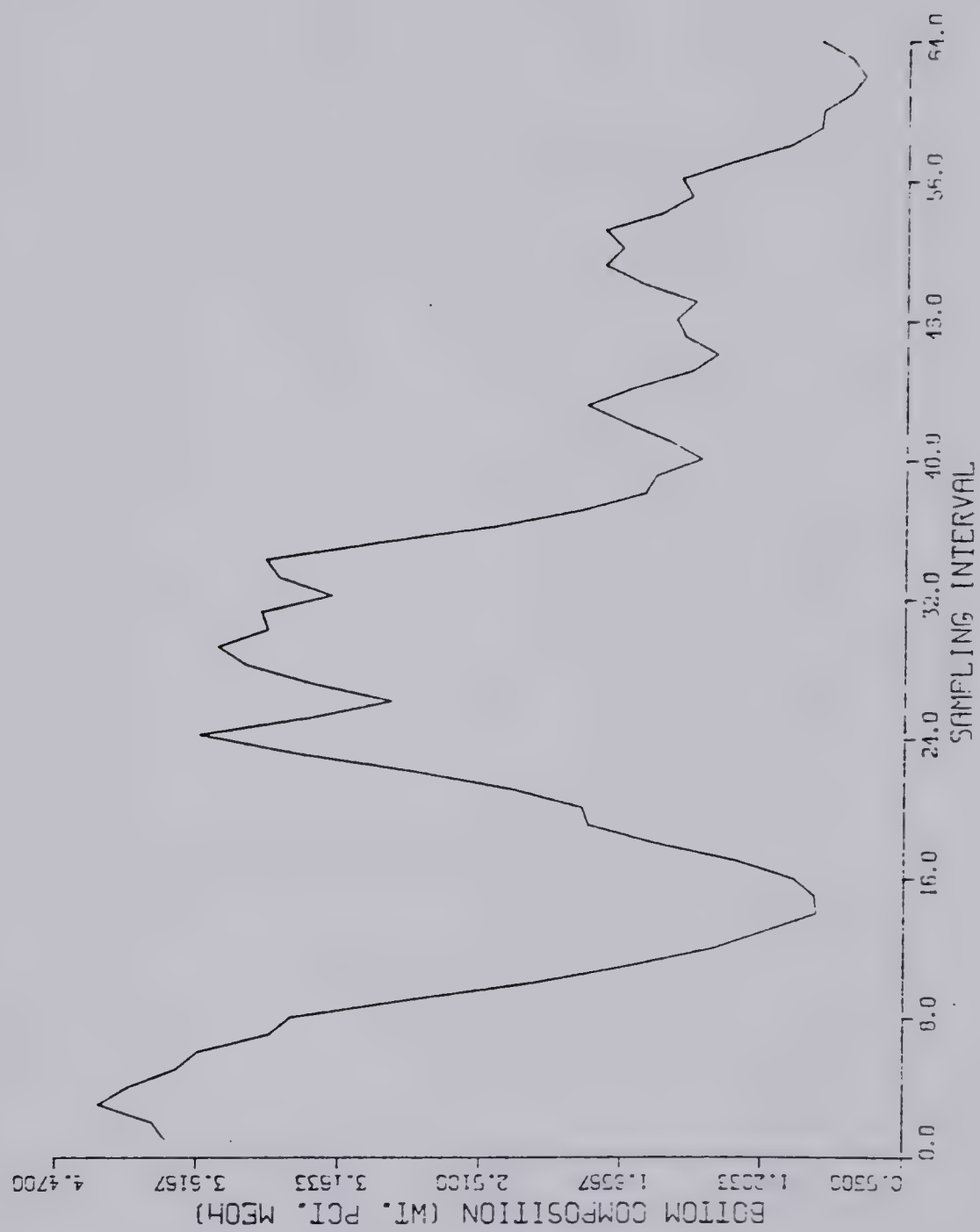


Figure C-7 Bottom Composition Response for PRBS
Disturbance of Length Double 31 using Steam

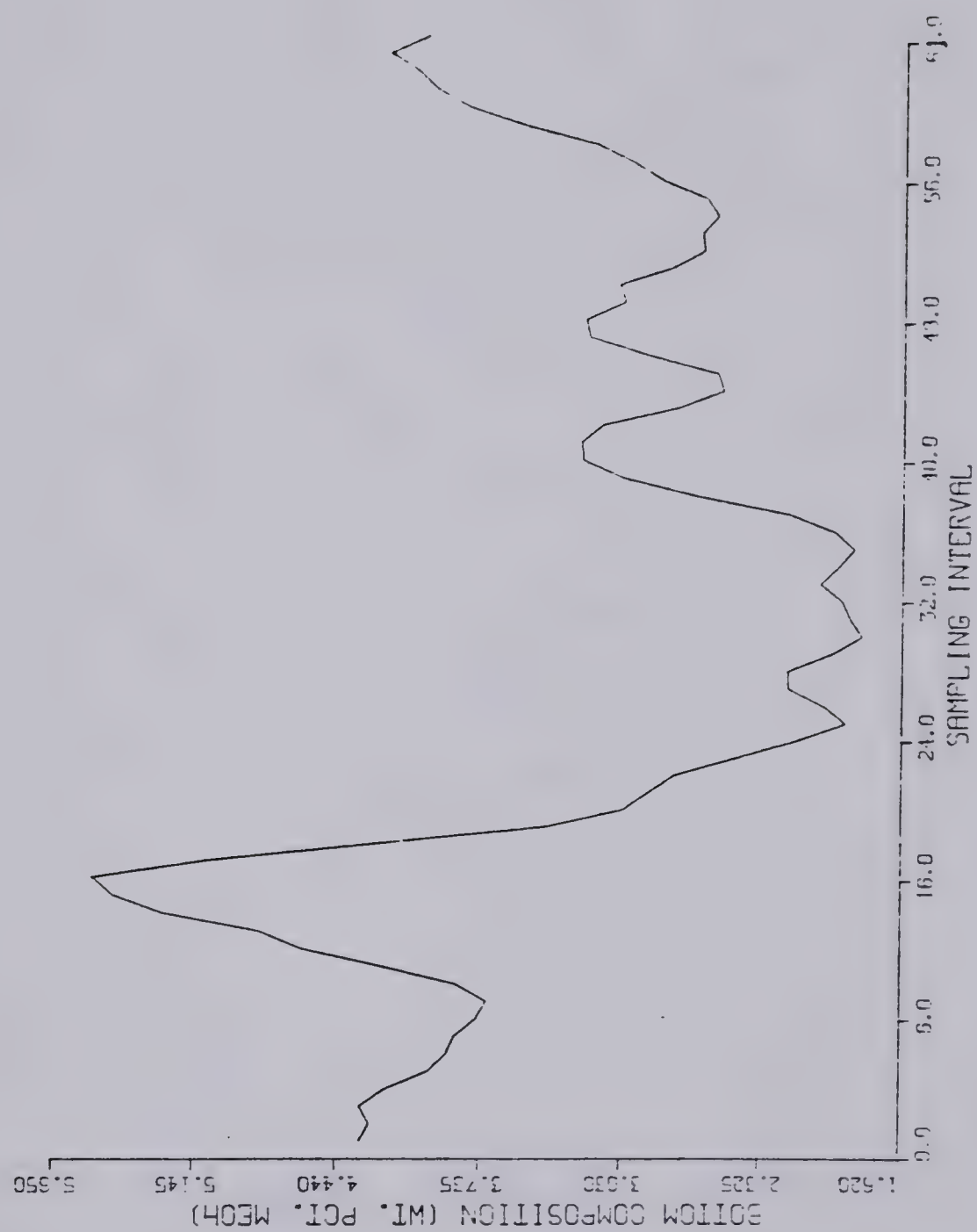


Figure C-8 Bottom Composition Response for PRBS
Disturbance of Length Double 31 using Reflux

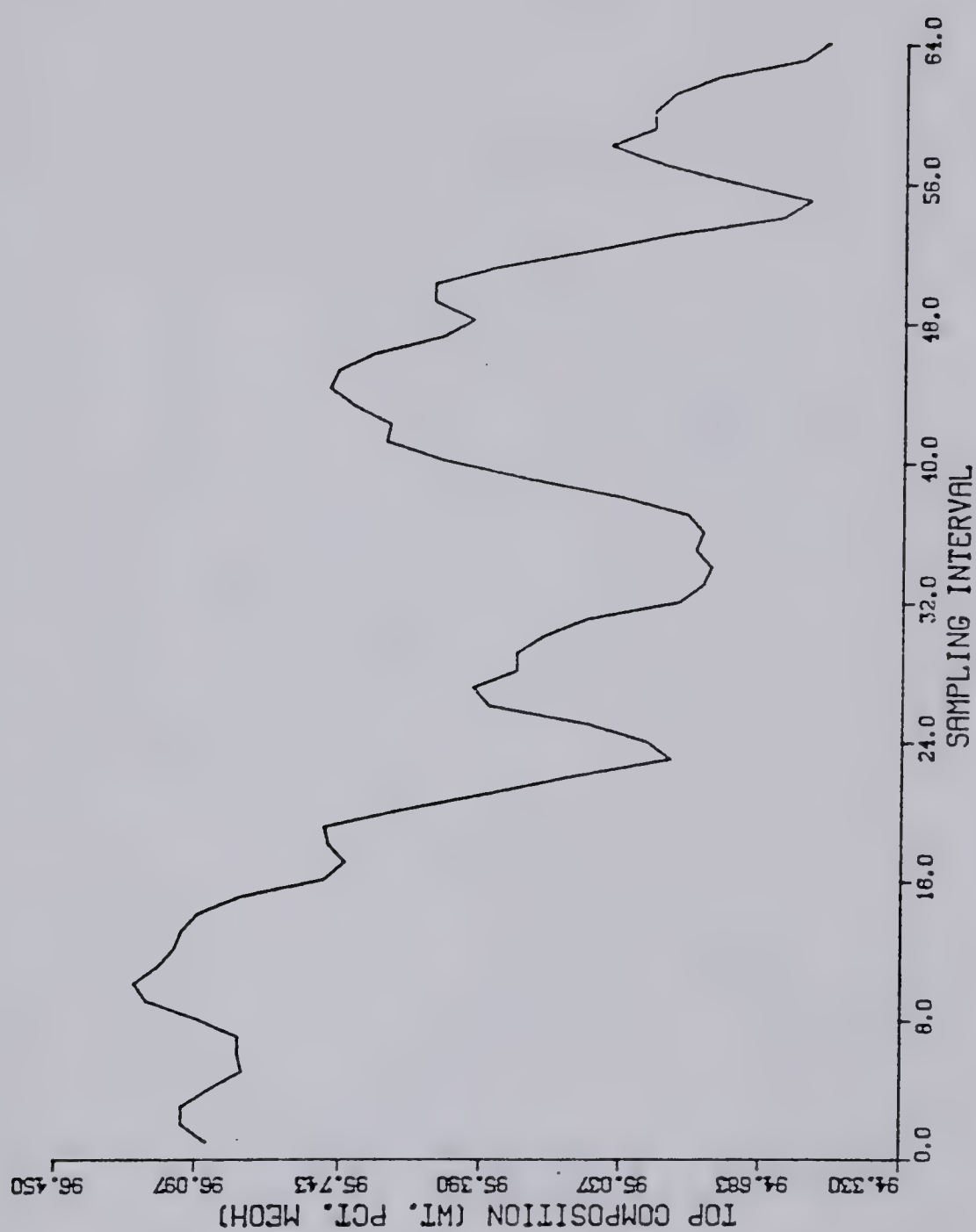


Figure C-9 Top Composition Response for PRBS
Disturbance of Length 63 using Steam

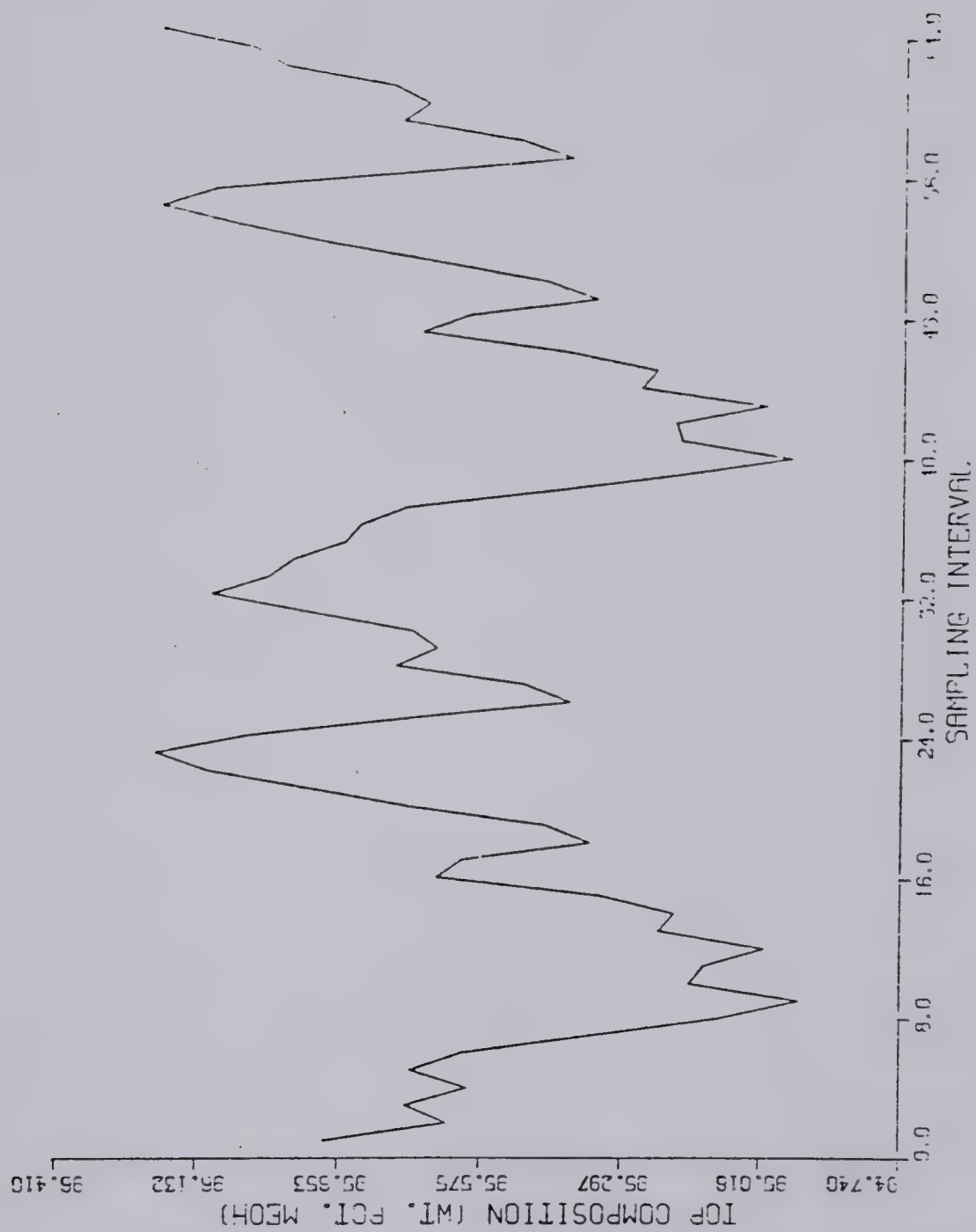


Figure C-10 Top Composition Response for PRBS
Disturbance of Length 63 using Reflux

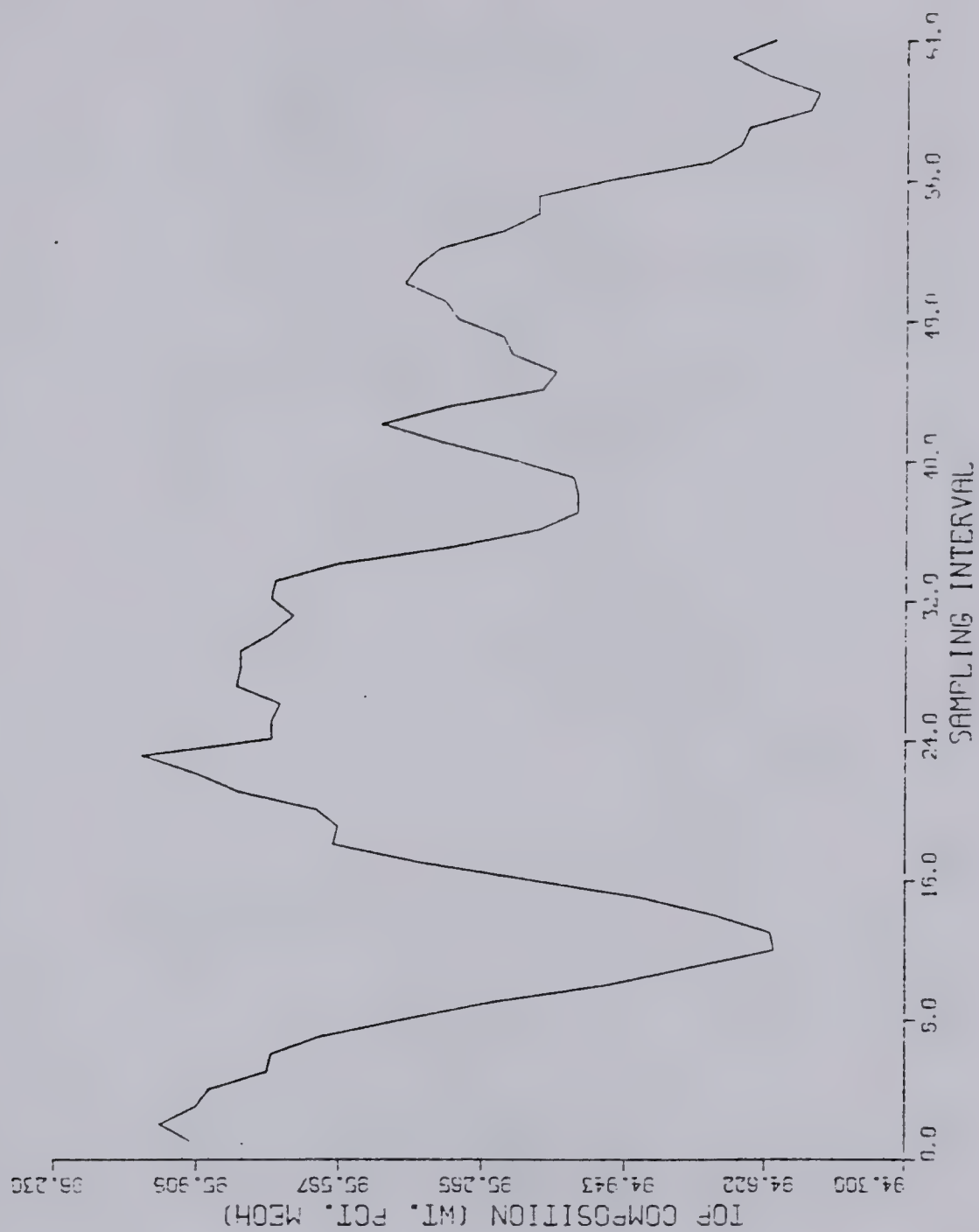


Figure C-11 Top Composition Response for PRBS
Disturbance of Length Double 31 using
Steam

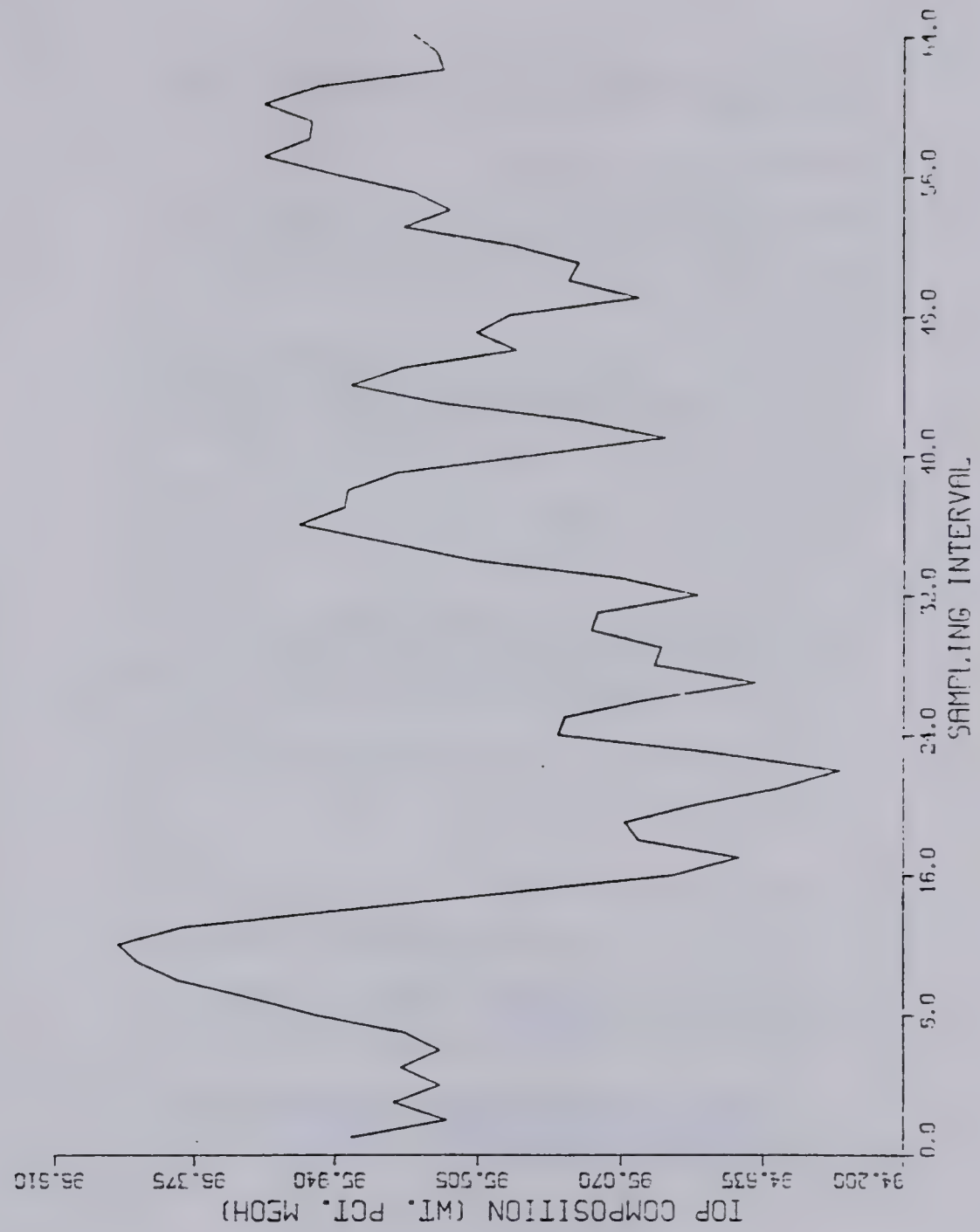


Figure C-12 Top Composition Response for PRBS
Disturbance of Length Double 31 using
Reflux



Figure C-13 Reflux Flow Rate Response for PRBS
Disturbance of Length 127

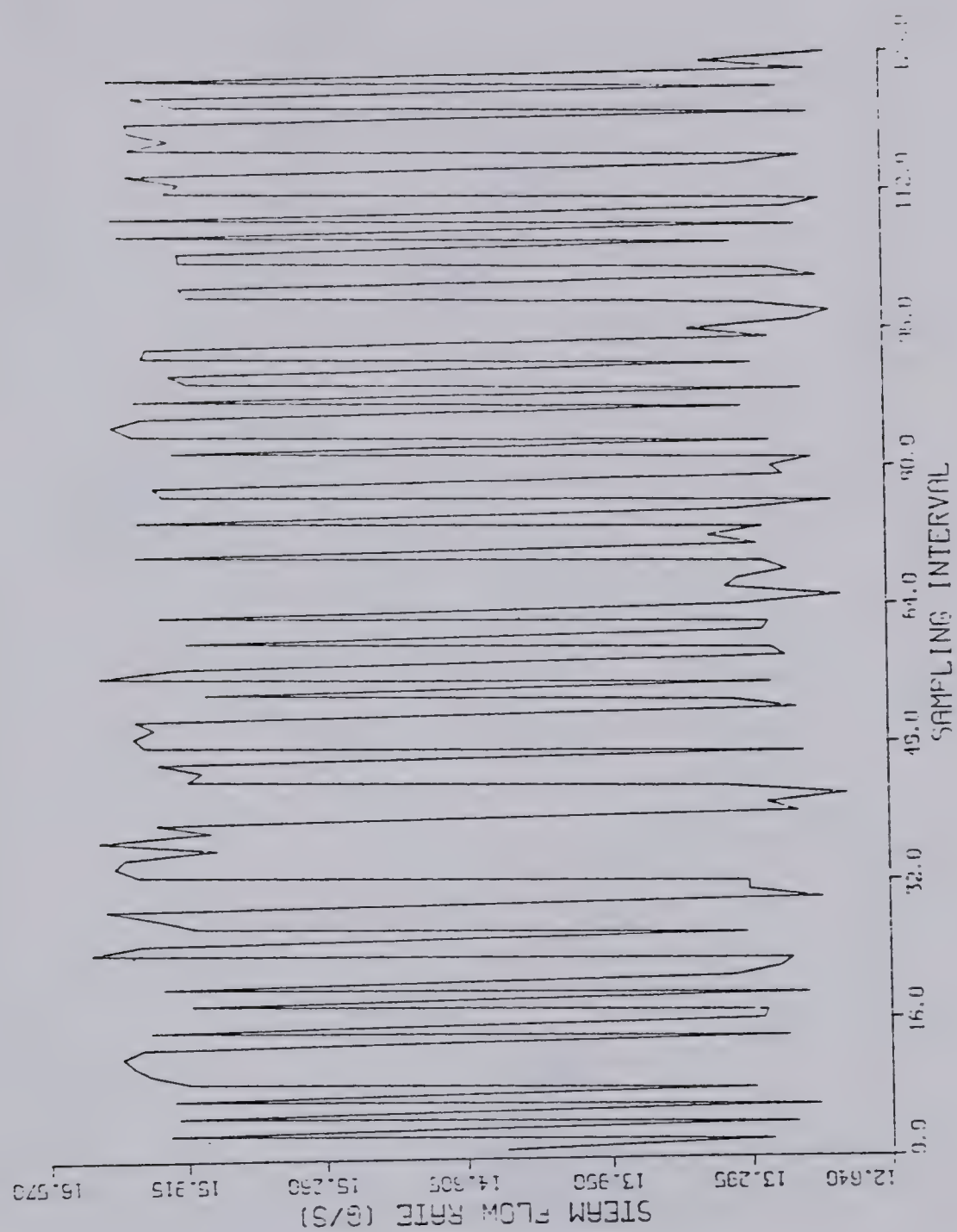


Figure C-14 Steam Flow Rate Response for PRBS
Disturbance of Length 127

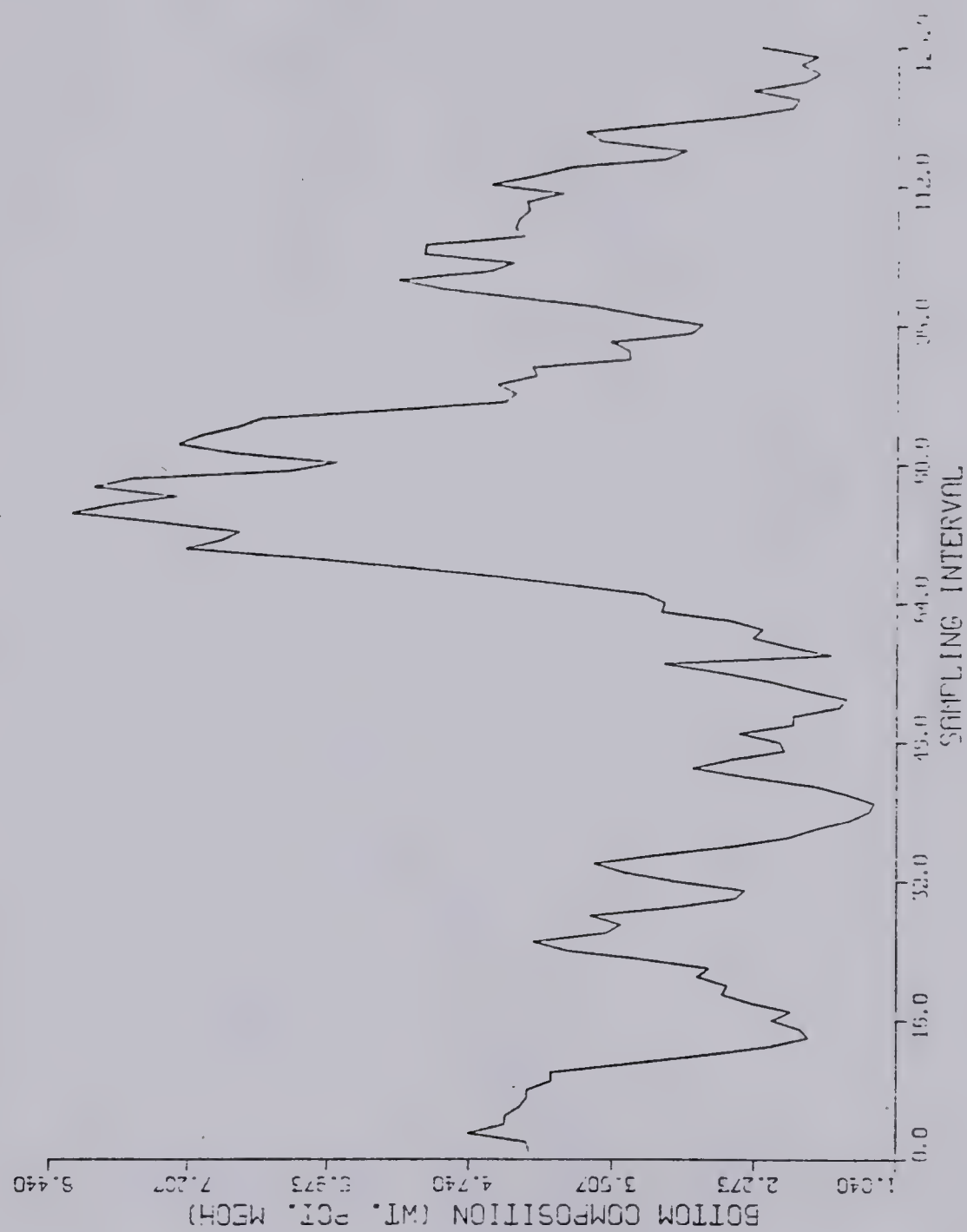


Figure C-15 Bottom Composition Response for PRBS Disturbance of Length 127 using Steam

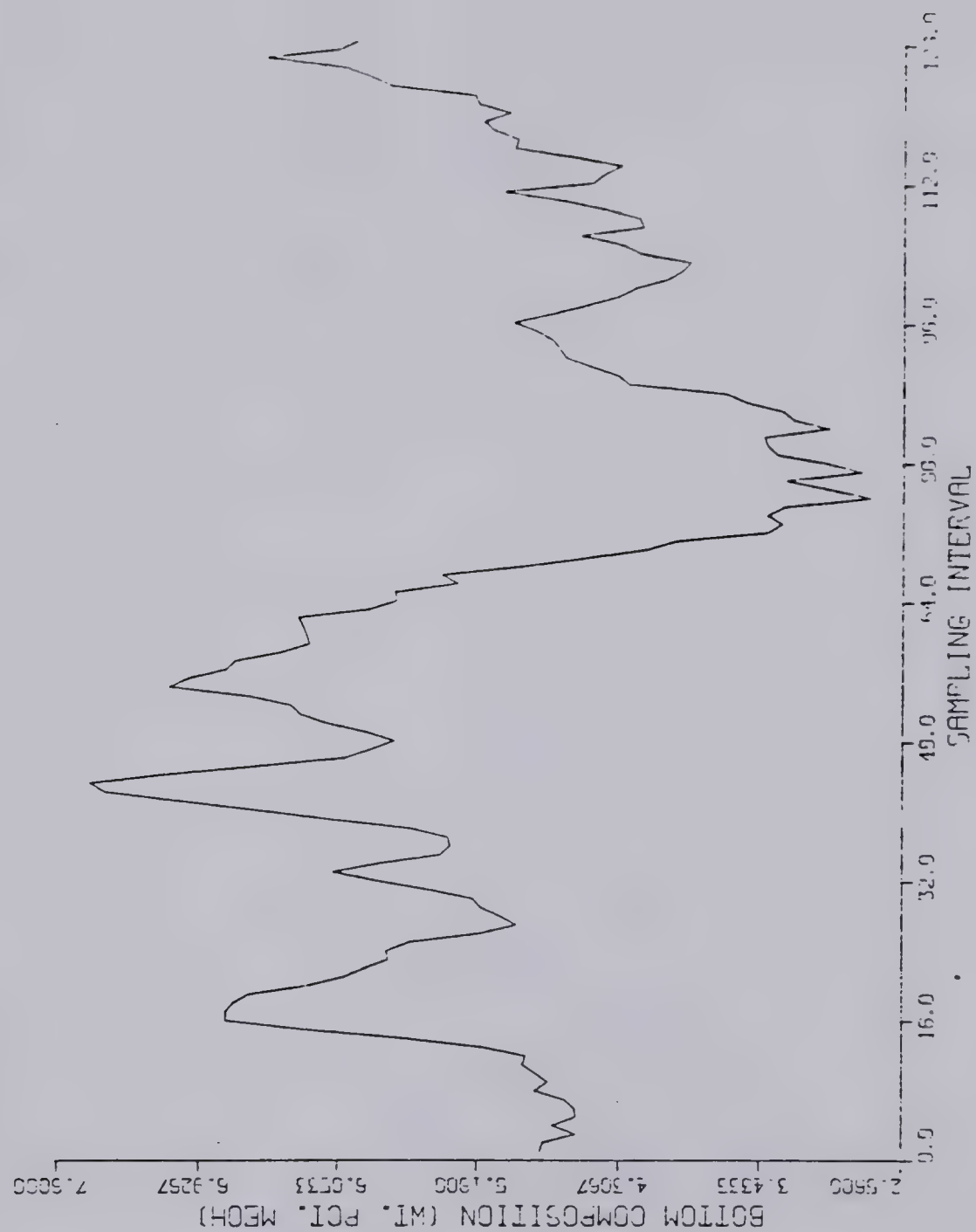


Figure C-16 Bottom Composition Response for PRBS
Disturbance of Length 127 using Reflux

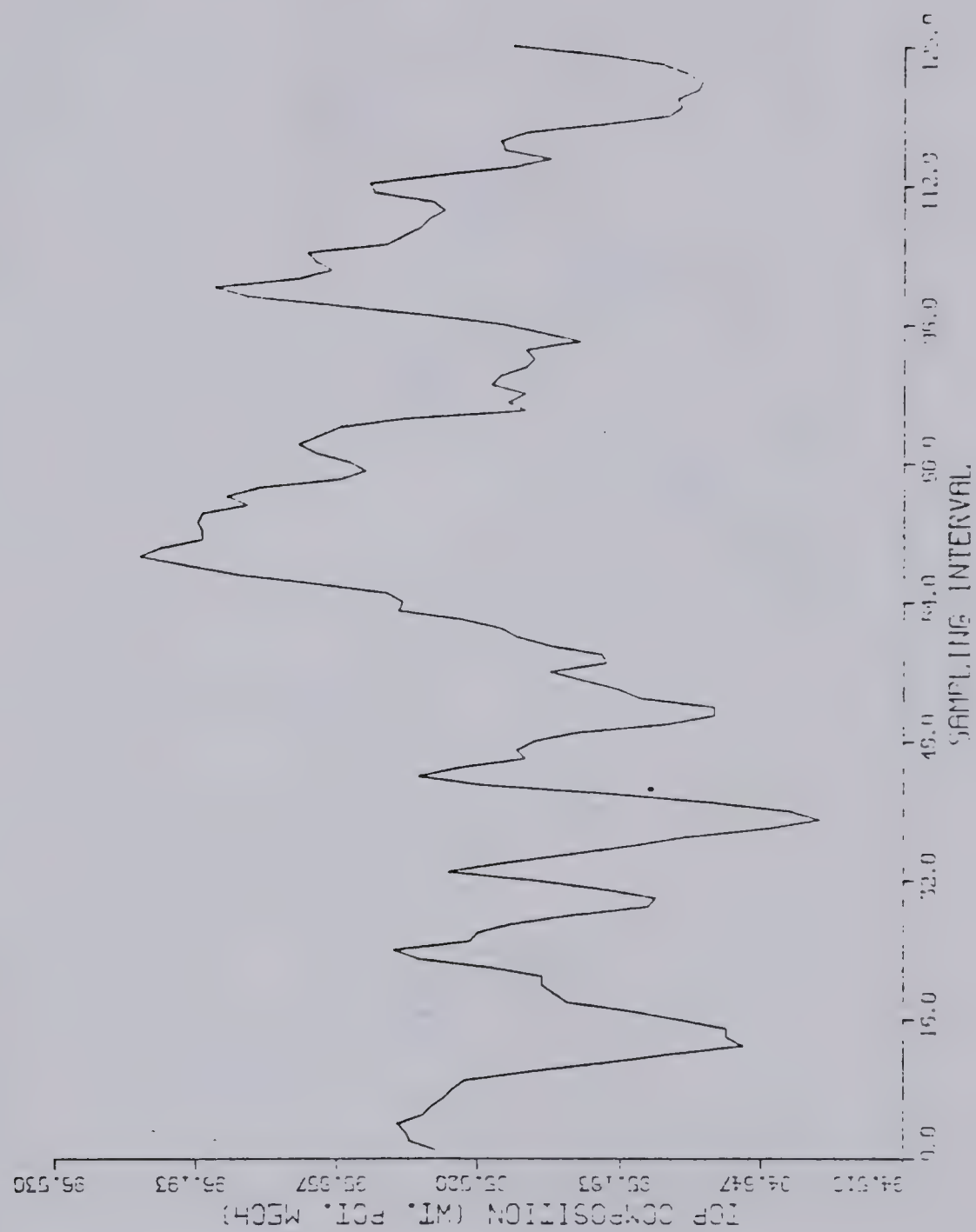


Figure C-17 Top Composition Response for PRBS
Disturbance of Length 127 using Steam

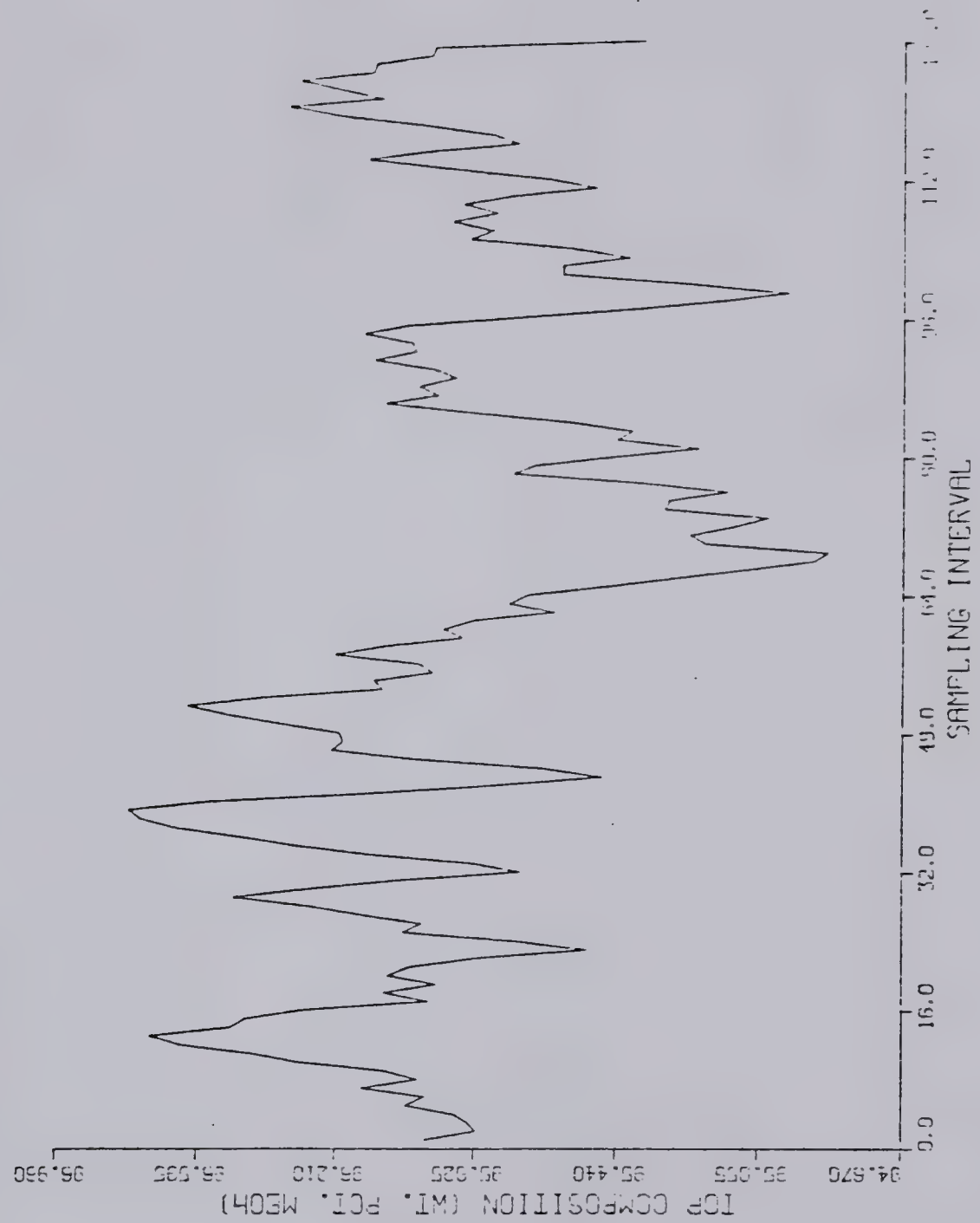


Figure C-18 Top Composition Reponse for PRBS
Disturbance of Length 127 using Reflux

APPENDIX D

GUIDELINES FOR PERFORMING PSEUDO RANDOM BINARY SEQUENCE IDENTIFICATION

This appendix contains a collection of guidelines developed during the testing of the two pilot units investigated during this thesis project that made identification of process units more systematic and less time consuming. The guidelines pertain only to open loop systems since only this type of system was investigated.

A) INPUT SIGNAL CHARACTERISTICS:

1) amplitude of input signal - this factor can be established by running open loop tests to determine system output response to step changes, up and down, from normal input conditions. The objective is to produce an output response that covers as much of the high and low ends of the spectrum of responses as possible without running into physical limitations on the system or operating constraints. UNALIP assumes the systems to be identified are linear. Non-linear behaviour becomes more apparent as the extremes of system response are reached and hence the validity of the linearity assumption can be checked more readily.

2) length of input signal - the length of the input signal depends on the dynamic behaviour of the system under consideration and the sampling interval. An observation made during the testing of the distillation column was that the length of the input signal should be at least as long as the largest time constant of the system under consideration. The step response of a first order system is only 63% complete after one time constant. Thus shortening the length of the input signal would omit a significant amount of response information. Sometimes system considerations or computer limitations do not make this possible. If this is the case then the general rule is that the longer the input signal length, the better the estimation. Some

work on sampling theory is discussed by Wozencraft and Jacobs [10] and can be applied in this area but was not done in this thesis.

3) sampling interval - the sampling interval is crucial to obtaining good system estimates and depends on system dynamics and physical limitations of the sampling mechanism. The sampling interval should be one-tenth as long as the smallest time constant of the system. When in doubt, the analysis can be repeated with several trial sampling intervals as a simple sensitivity test for the length of sampling interval. In the choice of sampling interval it is the output noise that is important, and its variance should approach a minimum value as the interval is shortened. Another guide for sample interval values is experiments done on similar equipment, as was the case with the heat exchanger experiments carried out in this thesis. The tradeoff of input sequence length vs sampling interval occurs if the total number of points is fixed or if large differences in system time constants exist. Thus the lower limit of a sampling interval for a system depends on system dynamics or computer storage size.

4) clock interval - the clock interval of a psuedo random binary sequence will depend on the size of the sampling interval and the magnitude of the system time constants. If the sampling interval is small and some of the time constants are large, then the clock interval

will have to be large to allow the system to 'see' the input disturbances. This will improve the estimates of the system gain which can be checked by doing open loop gain estimations.

B) SYSTEM CHARACTERISTICS:

1) model order - an initial estimate of the model order of a system can be determined from the characteristic shapes of the open loop response of the system to step changes in the input signal. Tests carried out on similar systems can also be used as a starting point. UNALIP offers a test to determine which model order is correct. If there is no dependence between the prediction errors and the input, then the assumed model order is adequate. The best model is determined by comparing the mean square error of different model orders. If the higher order model is significantly better than the lower order model, the F-ratio test will indicate this. If the covariance matrix has a determinant of zero then the model order is too high. For good estimates of the system model order, it is best to use several estimation routines to check on the results as was observed in the tests on the heat exchanger.

2) time delays - as with model order, initial estimates of the time delay can be obtained from open

loop experiments and experience with similar systems. UNALIP also has tests to determine if the time delay is correct. If there is no dependence between the residuals and the input then the time delay is correct. For time delays it is best to use more than one estimation routine to check the results, as was observed in the tests on the heat exchanger.

C) PROBLEMS:

1) noise - the higher the noise/signal ratio, the more difficult estimation becomes. The order of a system determines the amount of noise that can be part of a system's output before parameter estimation problems occur. The simulated first order system, with a noise/signal ratio of 5.0 had similar estimation success to a simulated second order system with a noise/signal ratio of 0.1. The higher the noise/signal ratio, the higher the failure of the estimation routines to converge or agree with other estimation routines.

2) model order - the larger the model order, the more difficult estimation becomes. Anything larger than a second order system significantly increased the calculation time required and resulted in models that gave good estimates of steady state gain but poor estimates for individual parameters.

3) time delays - the smaller the sampling interval, the more difficult time delay estimation becomes. This problem was noted in the tests done on the heat exchanger. This point should be remembered when placing confidence in time delay estimation of systems with fast dynamics, although as time delays become smaller their relative importance probably decreases.

B30285